



CLIENT GUIDE

RAIN OS documentation

Salesforce-native agents, communications, phone, outreach, insights, suggestions, scoring, administration, and integration guidance.

Contents

1. Start with RAIN OS
2. Setup Wizard and launch readiness
3. RAIN Hub, Inbox, and daily operations
4. Agent Builder, types, and use cases
5. Prompt Manager and Prompt Assistant
6. Context, variables, and object mapping
7. Test Environment and release validation
8. Phone numbers and add-ons
9. Outreach Lists and next-best actions
10. Voice, call, and chat settings
11. Agent assignment rules
12. Administration, permissions, and security
13. Agents, versions, and publishing
14. Automation and Flow actions
15. Communications and providers
16. Insights, suggestions, and scoring
17. Install, upgrade, and initialize RAIN OS
18. Operations and troubleshooting
19. Outreach and next-best actions
20. RAIN OS overview and boundaries
21. Phone, voice, and post-call processing
22. RAIN Intelligence compatibility
23. Reports, dashboard, and observability
24. Agent tools, knowledge, and actions
25. UI surfaces and navigation

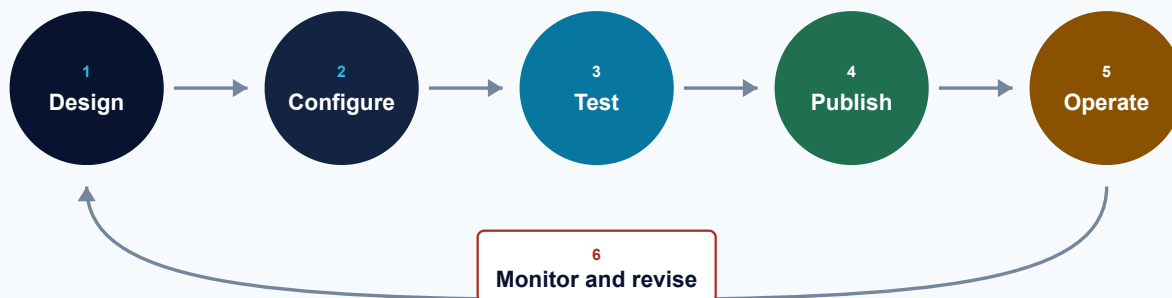
Start with RAIN OS

Choose the right RAIN OS learning path, understand the product boundary, and move from installation through safe production operation.

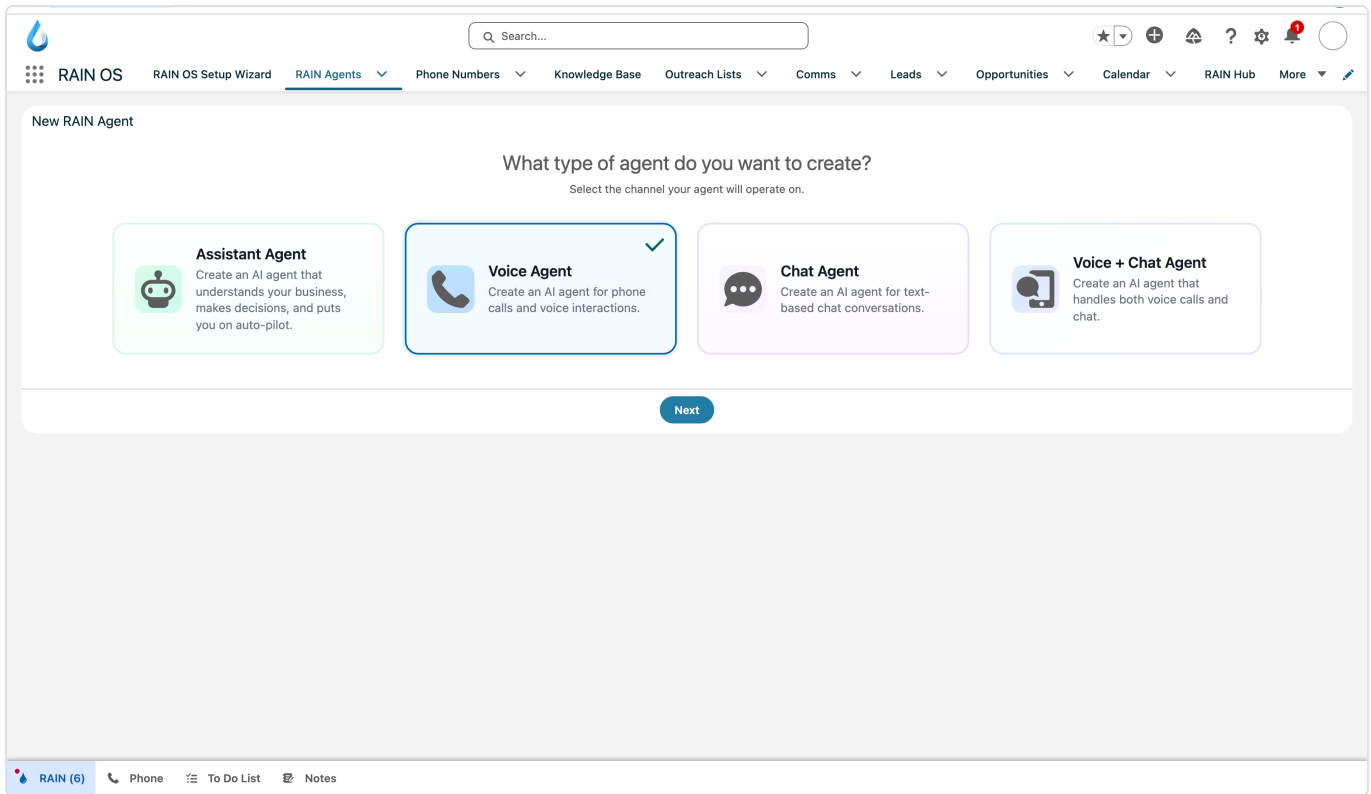
RAIN OS is a Salesforce-native foundation for agents, communications, phone and voice, outreach, recommendations, and operational AI. It can be installed and used independently. **RAIN Intelligence** is an optional add-on that extends reasoning, retrieval, tools, artifacts, and hosted workspaces.

From agent idea to controlled operation

Every production change returns through testing and review.



The operating lifecycle is to configure a bounded use case, test it, publish deliberately, monitor outcomes, and revise through a new version.



The Salesforce app navigation exposes Setup Wizard, RAIN Agents, Phone Numbers, Knowledge Base, Outreach Lists, Communications, business records, and RAIN Hub. The selected page is the safe first step of agent creation.

Choose a learning path

I need to...	Start here
Install or upgrade the package	Install, upgrade, and initialize RAIN OS
Complete org setup	Setup Wizard and launch readiness
Work daily recommendations	RAIN Hub, Inbox, and daily operations
Select an agent	Agent types and use cases
Write or improve instructions	Prompt Manager and Prompt Assistant
Map Salesforce data	Context, variables, and object mapping
Add knowledge and actions	Agent tools, knowledge, and actions
Test a voice or coach experience	Test Environment and release validation

I need to...	Start here
Publish a version	Agents, versions, and publishing
Acquire numbers or add-ons	Phone numbers and add-ons
Configure call or chat behavior	Voice, call, and chat settings
Run outreach or NBAs	Outreach Lists and next-best actions
Route records to an agent	Agent assignment rules
Connect Intelligence	RAIN Intelligence compatibility
Diagnose a problem	Operations and troubleshooting

Use one operating pattern

1. Define the business outcome, audience, authorized data, allowed actions, and human owner.
2. Configure the smallest agent and channel that can accomplish the job.
3. Add only necessary context, knowledge, variables, and tools.
4. Test normal, incomplete, restricted, and failure cases.
5. Publish a traceable version and release it to a controlled audience.
6. Monitor Salesforce records and provider results separately.
7. Expand only after completion and recovery behavior are reliable.

IMPORTANT

AI output, provider acceptance, and a successful Salesforce save are different stages. Verify the authoritative record and final provider status before treating work as complete.

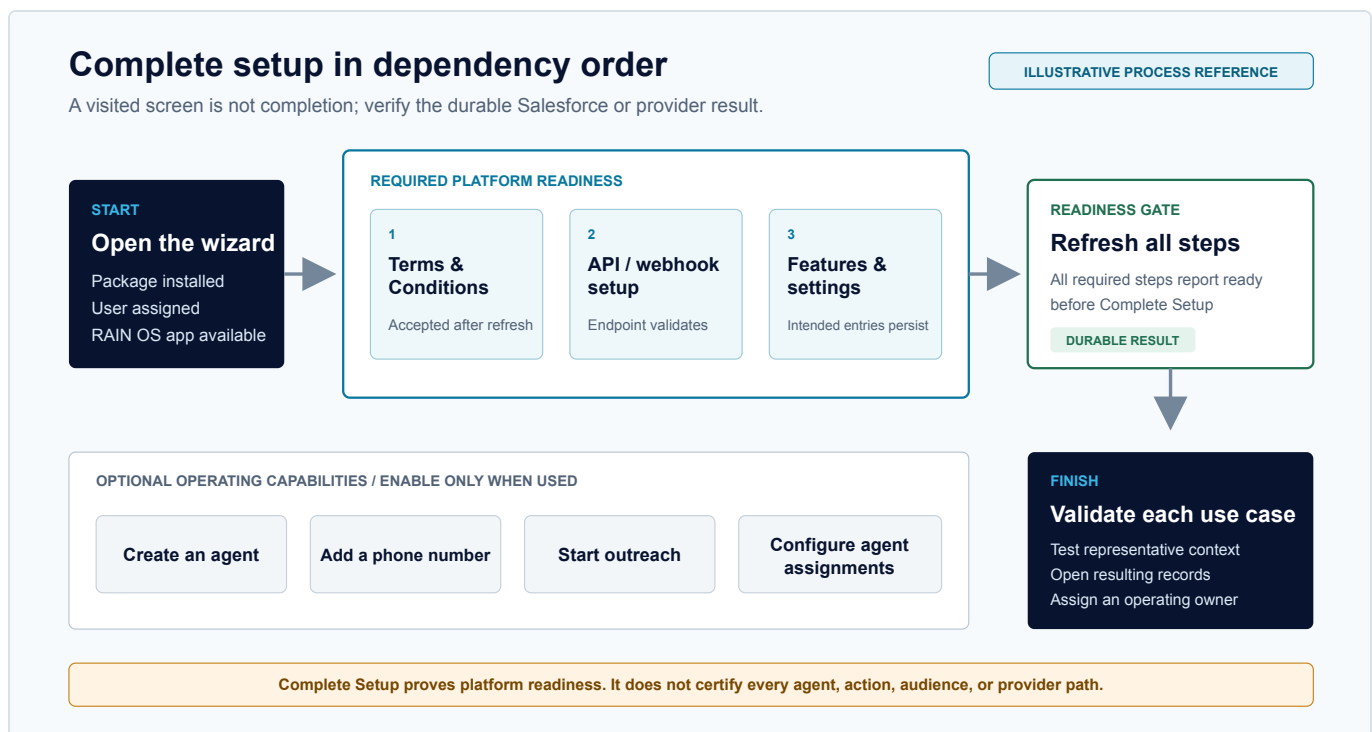
Setup Wizard and launch readiness

Complete required RAIN OS setup in dependency order, then configure agent, phone, outreach, and assignment steps only for enabled use cases.

Open the Setup Wizard

Open the **RAIN OS** app, then select **RAIN Setup Wizard**. The wizard is both a setup path and a readiness checklist. A green or completed step should reflect a durable Salesforce or provider result, not only that a user visited the screen.

Package installation and org setup are separate jobs. Complete Install, upgrade, and initialize RAIN OS before using this wizard. Salesforce also documents how to assign packaged permission sets [🔗](#); assignment still depends on the user's license.



Complete platform prerequisites and required readiness checks before enabling optional capabilities, then validate one bounded use case end to end.

Complete required setup first

Step	Purpose	Completion evidence
Terms & Conditions	Accept the required compliance terms before production use	The wizard reports the terms as accepted after refresh
API/Webhook Setup	Establish the public-facing endpoint required for external communication callbacks	The configured site and callback path validate successfully
Features & Settings	Enable and configure only the RAIN capabilities intended for the organization	Settings persist and the intended application entries are available

Use **Refresh All Steps** after completing a setup action outside the current wizard screen.

Complete Setup should be used only after every required step reports ready. Do not treat a visited step, saved screen, or successful package install as proof that a site, credential, callback, or provider is usable.

Add optional operating capabilities

After required setup, complete only the operational steps required by the enabled use cases:

1. **Create an Agent** when an enabled use case uses a RAIN agent, then open the new record to configure, test, and publish it.
2. **Add a Phone Number** when an enabled use case uses the voice channel, then purchase or synchronize an eligible number and map it to the intended agent and version.
3. **Start Outreach** when an enabled use case uses Voice Calls or Next Best Actions, after a published agent and linked phone number are ready.
4. **Agent Assignments** when an enabled use case selects a published prompt from supported record context.

These four steps are optional for base platform readiness but become required for any enabled use case that depends on them. Outreach remains locked until its displayed readiness prerequisites are met, and optional-step availability can vary with the enabled feature set.

Launch checklist

- The intended users have the RAIN OS app, tab, object, field, Apex, component, and record access required for their jobs.

- Required terms, site, webhook, service URL, and provider configuration are complete.
- For each enabled agent-backed use case, at least one low-risk agent is configured, tested with representative context, and published.
- For each enabled phone or messaging channel, the provider account and sender are approved, available, and mapped.
- For each enabled outreach or autonomous use case, quiet hours, consent, suppression, approval, and action policies are documented.
- For each enabled assignment use case, record eligibility, selection priority, and fallback ownership are documented and tested.
- For each enabled outreach or suggestion use case, a controlled test produced the expected Salesforce and provider results.
- Owners know where to review Communications, RAIN Hub, Error Log, reports, scheduled work, and provider status.

Finish by recording the owner and validation evidence for each enabled use case. Launch readiness is complete only when an operator can open the expected Salesforce records, identify the configured provider path, and explain where failures will be reviewed.

IMPORTANT

Completing the wizard enables the configured platform path. It does not certify every agent, tool, audience, phone number, assignment, or autonomous action. Validate each production use case separately.

If setup will not complete

Check the step's actual record or endpoint, the running user's permission, setup status refresh, service banner, browser errors, and Salesforce error logs. When outreach remains locked, verify a published eligible agent version and linked phone number before recreating either record. When a tab is missing, distinguish permission assignment from app-navigation enablement.

RAIN Hub, Inbox, and daily operations

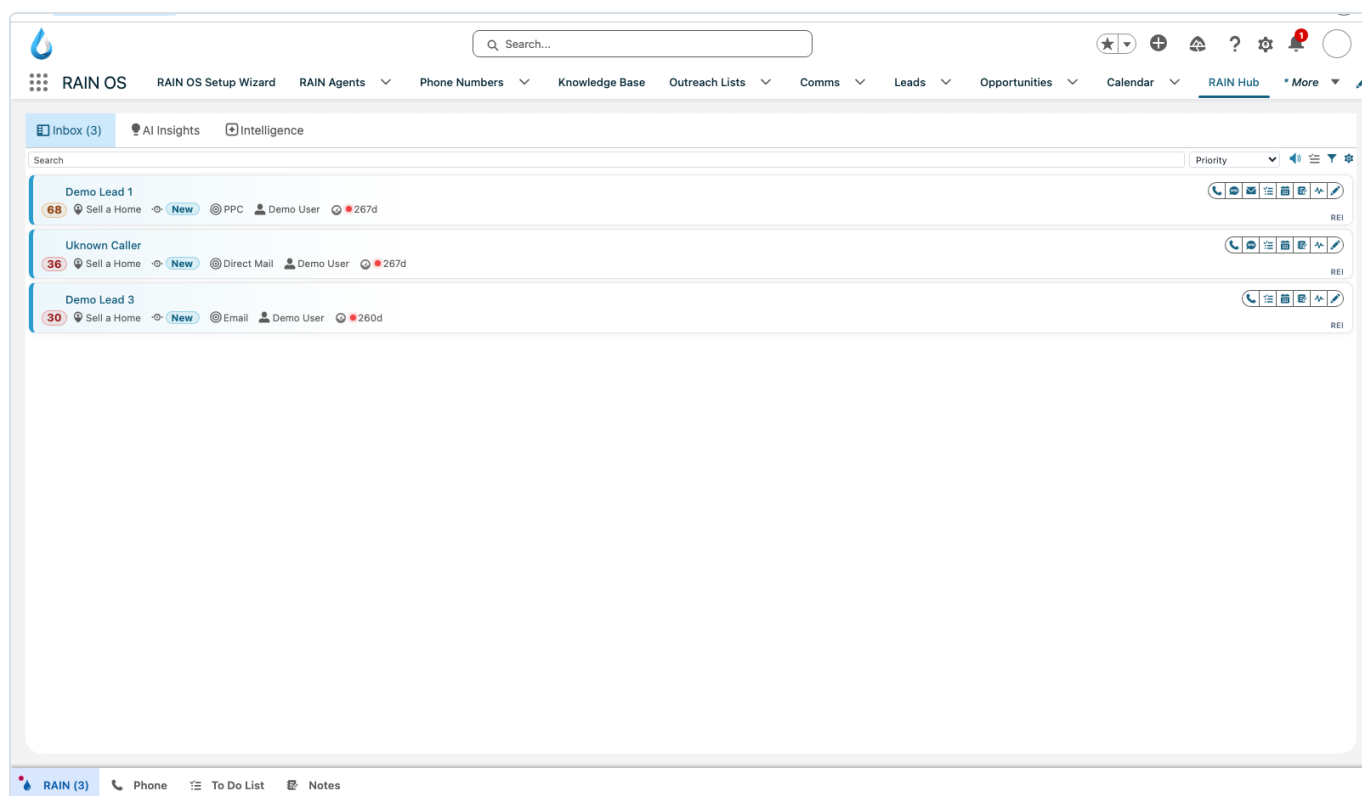
Operate Inbox, AI Insights, and Intelligence from RAIN Hub, and verify each recommendation or communication against its authoritative Salesforce record.

Find RAIN Hub

RAIN Hub is available as a RAIN OS application tab and as a utility item when assigned. It presents:

- **Inbox**, the default attention and work queue.
- **AI Insights**, generated observations and recommendations.
- **Intelligence**, the optional hosted reasoning and workspace experience.

The utility label and highlight can update when new Inbox or insight events arrive.



The synthetic Inbox shows the daily triage controls together: work count, search, priority sort, display and filter controls, score and status chips, ownership, age, and row actions.

Work the Inbox

The current Inbox attention queue is centered on eligible Leads and Opportunities. Use search, sort, filters, and configured display chips to narrow the list. Results are paged, so clear filters before deciding an item is absent. Open the underlying Salesforce record before acting.

Available row actions depend on the record, item state, enabled features, and user access. They can include claiming work where the claim control is shown, marking an item read, opening the record, calling, sending SMS or email, creating a Task, Event, Note, or Chatter post, and reviewing or clearing a recommended action.

Mark All Read acknowledges attention items; it does not complete their underlying work. Inbox Display Chips control which selected Lead or Opportunity values appear on cards and can be customized without changing the business records.

Review AI Insights and suggestions

Evaluate the evidence, target, proposed action or value, confidence, freshness, ownership, and permission boundary. Act only when the current record still supports the recommendation. After an action, verify the resulting Salesforce record or field. Reject or clear stale, unsafe, incorrect, or unnecessary work as the interface allows.

Salesforce object, field, and record access still determine what a user can open or change. The Salesforce data security model  explains why a visible Inbox item does not itself grant access to the related record.

Understand completion

An Inbox action is complete when the intended durable result exists and responsibility is clear. A submitted provider action can still be queued. A read item can still have an open Task. A suggestion can show approval while a later provider or background action is pending. Follow the chain through Salesforce status, queued work, provider result, and reconciliation.

Troubleshooting

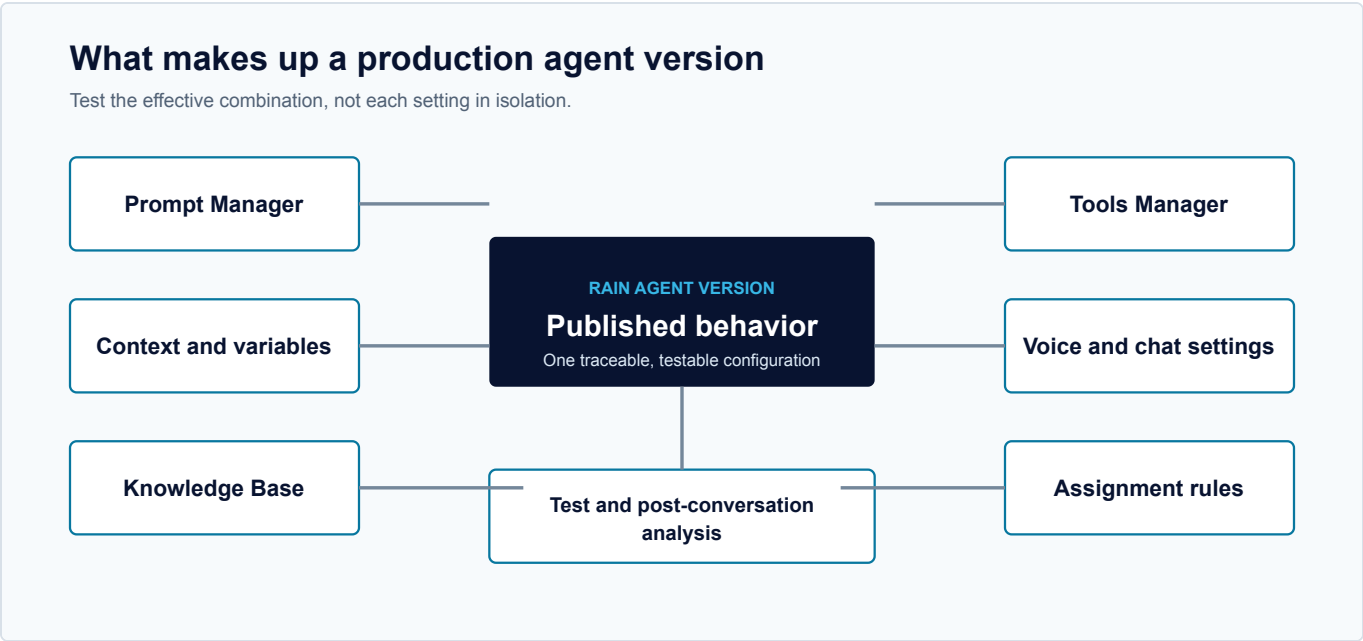
For missing items, confirm first that the record is an eligible Lead or Opportunity, then review filters, read state, ownership, queue or team eligibility, record sharing, the event source, and background processing. For a mismatched count, clear search and filters and account for pagination and event timing. For an inaccessible related record, check sharing separately from RAIN-item access and use

User Access Summary  to trace assigned access. For missing sound, check the Inbox preference and browser media policy.

Agent Builder, types, and use cases

Choose the smallest capable agent type, create its durable identity, and use Agent Builder to configure a testable draft version for the intended channel.

Choose the smallest capable agent

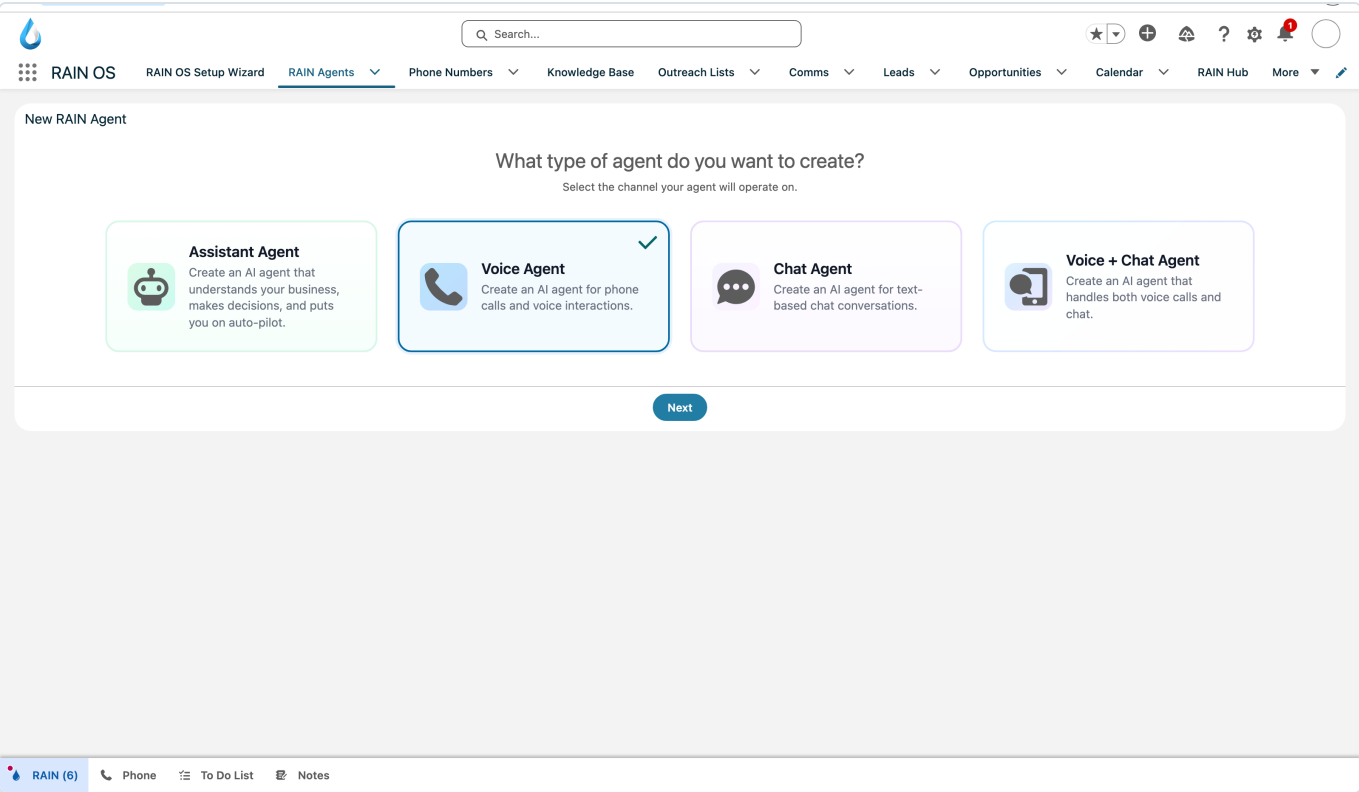


This map groups the configuration areas that combine into one testable agent version.

Agent type	Best fit	Configuration emphasis
Assistant	Salesforce-based assistance and record-assigned prompts	Assistant prompt, knowledge, assignment, permissions, and approval boundaries
Chat Agent	Text conversations through a connected chat experience	Prompt, context, tools, knowledge, chat inactivity, and escalation
Sales Coach	Practice conversations with AI playing the customer	Methodology, offering, customer profile, role-play direction, and coaching expectations
Voice Agent	Inbound or outbound phone conversations	Phone mapping, voice, speech, silence, voicemail, transfer, call limits, and post-call

Agent type	Best fit	Configuration emphasis
		extraction
Voice + Chat Agent	One designed behavior exposed through both supported channels	Shared prompt and knowledge plus separate channel-specific validation

Do not select a broader agent type solely for future possibility. Every enabled channel increases configuration, testing, monitoring, and compliance responsibility.



The first creation step makes the channel boundary explicit before any prompt, provider, number, knowledge, or tool configuration is entered.

Create the agent identity

From the Agents surface, choose **New Agent**, select one of the five types, and complete the fields shown for that type. Model, language, and voice requirements differ by type. Assistant creation does not require the voice fields used by channel agents, while Sales Coach also requires a methodology. Treat the creation form as authoritative for required values.

Choose the type carefully. The agent type controls which builder sections and channel settings are available. A conversion or clone can create a starting point for another channel, but every copied

prompt, variable, tool, knowledge source, and channel setting requires review.

Configure a draft in Agent Builder

Open the agent and select an editable draft version. The builder exposes **Details** and **Prompt Manager**, then the context, knowledge, tools, test, and channel sections supported by the type. Published or otherwise locked versions are reference points, not working drafts.

Use this order:

1. Confirm the agent type, owner, channel, language, and model choices.
2. Define prompt behavior and prohibited actions.
3. Add the minimum context, variables, knowledge, and tools.
4. Configure voice, call, or chat behavior when that channel is present.
5. Test the draft and correct validation errors.
6. Publish through the version workflow only after review.

Agent and version responsibilities

The Agent is the durable identity and type. An Agent Version contains the behavior that is tested and published: model, language, prompt, context, knowledge, tools, variables, channel settings, and post-conversation extraction. Preserve prior versions for traceability and publish intentionally. See Agents, versions, and publishing for the release workflow.

Define the use case before configuration

Write down:

- The audience and channel.
- The business outcome and authoritative Salesforce record.
- Information the agent may read.
- Actions the agent may take, recommend, or never take.
- Consent, quiet hours, disclosure, recording, transfer, and escalation requirements.
- Completion and failure signals.
- The human owner for exceptions.

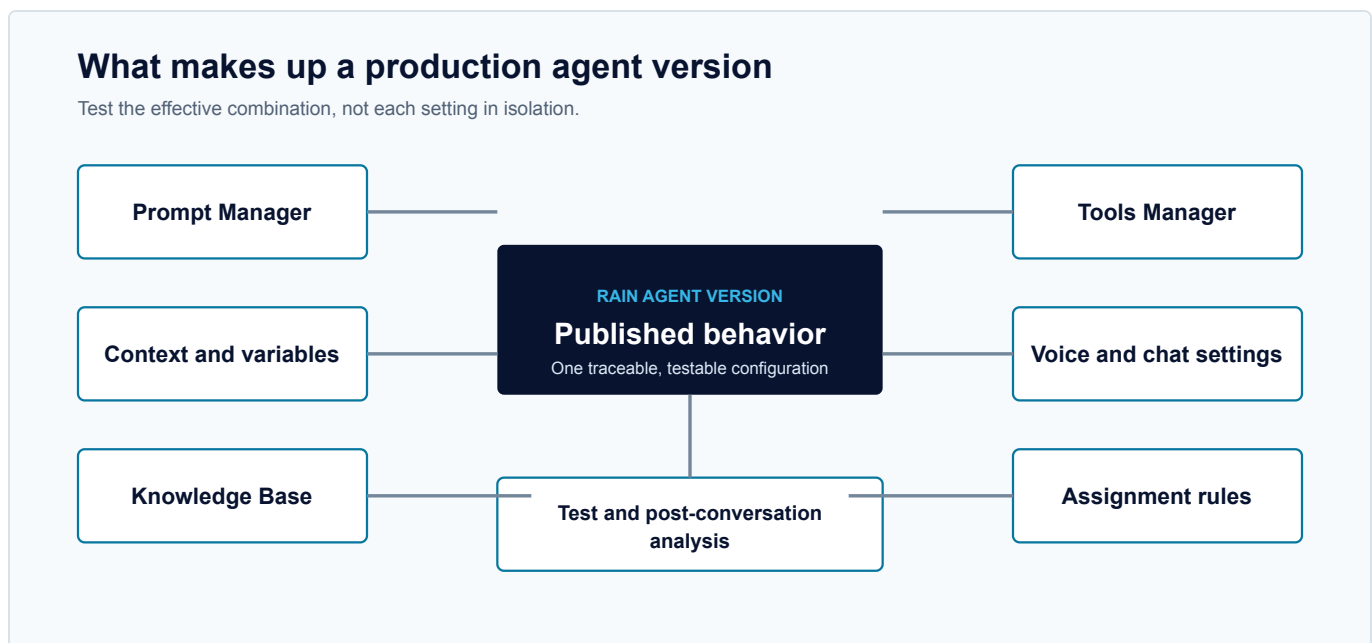
Readiness criteria

The selected type is appropriate when its required channel, context, tools, and safeguards are available. The agent is ready only after representative tests show correct behavior for normal, incomplete, ambiguous, adversarial, and provider-failure cases.

Prompt Manager and Prompt Assistant

Build clear agent instructions in Prompt Manager, use Prompt Assistant for reviewable section-level improvements, and test every accepted change before publishing.

Structure the prompt



This map places Prompt Manager and Prompt Assistant inside the complete version boundary from drafting through testing and activation.

Prompt Manager is the primary place to define agent behavior. Use explicit sections so people and Prompt Assistant can review one concern at a time:

1. Role and business purpose.
2. Audience, channel, and disclosure.
3. Allowed information and authoritative sources.
4. Conversation sequence and required discovery.
5. Tools and the conditions for using them.
6. Actions requiring confirmation or human approval.

7. Prohibited actions and safety boundaries.
8. Escalation, transfer, failure, and end conditions.
9. Output, data-capture, and completion expectations.

Avoid embedding volatile record values in the prompt. Use configured variables and record context. Avoid telling an agent that a tool succeeded before the tool returns a success result.

Prompt Manager supports a single prompt or a structured prompt with sections and subsections. Choose a structure that makes ownership and review clear; changing format is not a substitute for resolving conflicting instructions. Salesforce's Prompt Fundamentals [🔗](#) provides a platform-neutral introduction to goals, context, constraints, grounding, and iteration.

Use Prompt Assistant

Open **Prompt Assistant** from Prompt Manager when its service is connected and available. Choose **Analyze**, **Fix**, **Rewrite**, or **Write New**, then ask a specific question such as improving transfer rules, reducing ambiguity, or adding a missing refusal condition. The assistant can consider the current prompt, selected knowledge, and supported attachments, then return section-level additions, replacements, or removals.

Review each proposed block and its diff. Approve or reject suggestions individually; use bulk approval only after reviewing every change. Accepted suggestions update the working draft, but do not make that behavior production-ready without save, test, and publish steps.

The current interface limits a request to three attachments of up to 5 MB each, five history items, five returned suggestions, and eight selected knowledge sources. These are release limits; follow the interface when it reports a different value.

Prompt content, request history, selected knowledge, and attachments are sent to the configured AI service for processing. Processing can include provider-hosted response storage and web search when enabled. Do not attach secrets, credentials, unnecessary personal data, internal-only material, or content that the organization has not approved for that service.

IMPORTANT

Prompt Assistant accelerates drafting. The administrator remains responsible for correctness, privacy, safety, tool authorization, testing, and release approval.

Keep prompts maintainable

- Give one instruction a single clear owner and location.
- State precedence when rules can conflict.
- Use positive steps and explicit prohibitions where risk requires both.
- Refer to tools by their configured purpose and expected return, not an assumed implementation.
- Define what to do when context is blank, contradictory, stale, or unauthorized.
- Keep channel-specific voice or chat behavior out of unrelated sections.

Validate every change

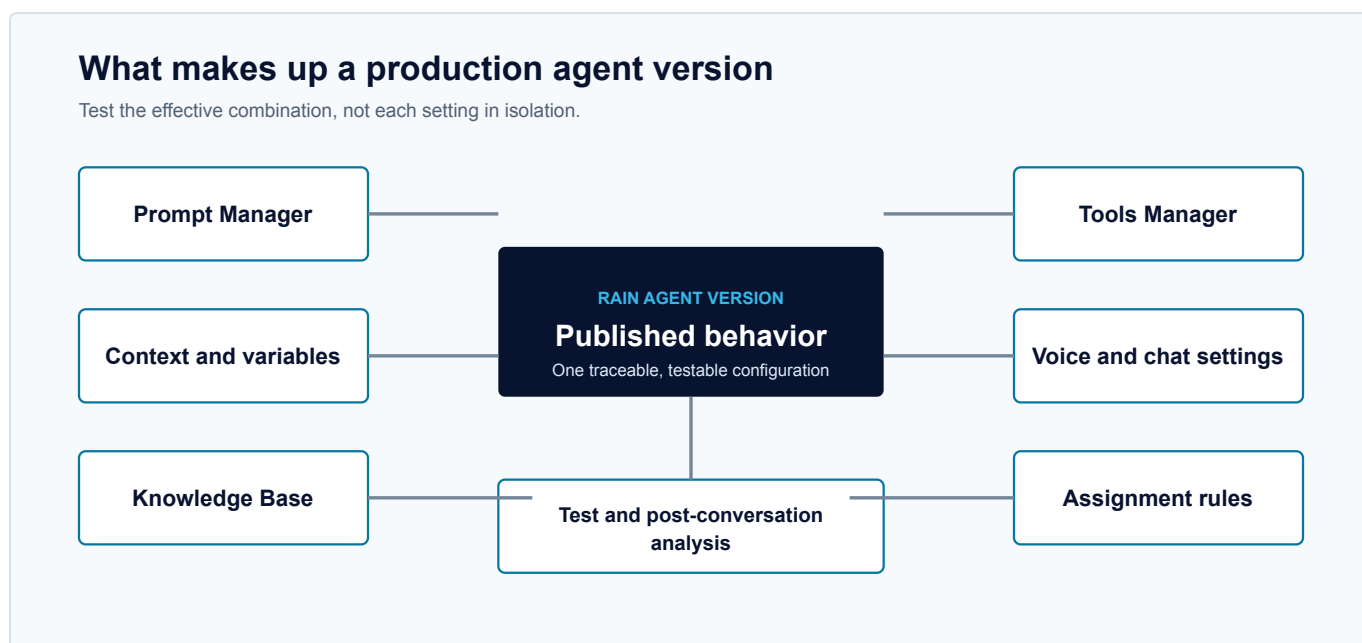
Save the draft version, test representative records and manual variables, test unavailable tools and missing context, review transcripts and extracted data, and publish only after approval. If a prompt change affects tools, context, assignment, or channels, include those components in the same regression test.

Salesforce's prompt planning guidance  reinforces iterative testing and human review. It does not document RAIN Prompt Manager behavior; use it for general prompt-design practice.

Context, variables, and object mapping

Select eligible Salesforce objects, map record data into named variables, preview effective context, and handle blanks and permissions before the agent uses the information.

Separate prompt, context, and variables



The configuration map shows context and variables as versioned inputs alongside prompts, knowledge, tools, channels, and testing. It does not represent the Context Manager UI.

- The **prompt** defines stable behavior and rules.
- **Context** provides approved record or business information.
- **Variables** provide named runtime values that the prompt, tools, or provider can reference.
- **Object mapping** determines which Salesforce record fields can populate those variables for an eligible object.

Keeping these responsibilities separate makes behavior easier to test and prevents volatile customer values from becoming permanent instructions.

Configure Context Manager

1. Select only the Salesforce objects required for the agent's use case.
2. Decide whether record detail and activity history are necessary.
3. Add variables with clear names, business meaning, type expectations, and safe defaults.
4. Map each supported object to the field that supplies each variable.
5. Define behavior for blank, inaccessible, or invalid values.
6. Preview the context using representative records from every selected object.

The current object-source choices are Lead, Account, Contact, Opportunity, and Case. A text variable accepts a default value. A picklist variable defines allowed options and a default. Each selected object can map the same variable token to the field appropriate for that object.

Context can include direction, session type, agent number, selected object and record, time zone and current time, record details, and activity history according to configuration. The Test Environment also allows manual variable values. Manual values do not prove that production object mapping works.

Design variables safely

Use a stable label, token name, and one meaning per variable. Do not place secrets, credentials, or unnecessary sensitive data in variables. Treat defaults as fallback context, not verified record facts. When a variable controls a tool or high-impact action, validate its allowed values before execution.

Test record mapping

Select a record in the Test Environment and review **Live Preview**. Confirm the intended values, labels, blanks, relationship traversal, date/time zone, and activity context. Repeat for records with sparse data, unexpected status, restricted sharing, and values near format limits.

Salesforce access remains part of the boundary, but the effective execution identity varies by path. Administrators must document the identity for each enabled channel and tool rather than assuming every read and write is equivalent to the interactive user. Use Salesforce field-level security guidance 🌈 when deciding which fields may be exposed.

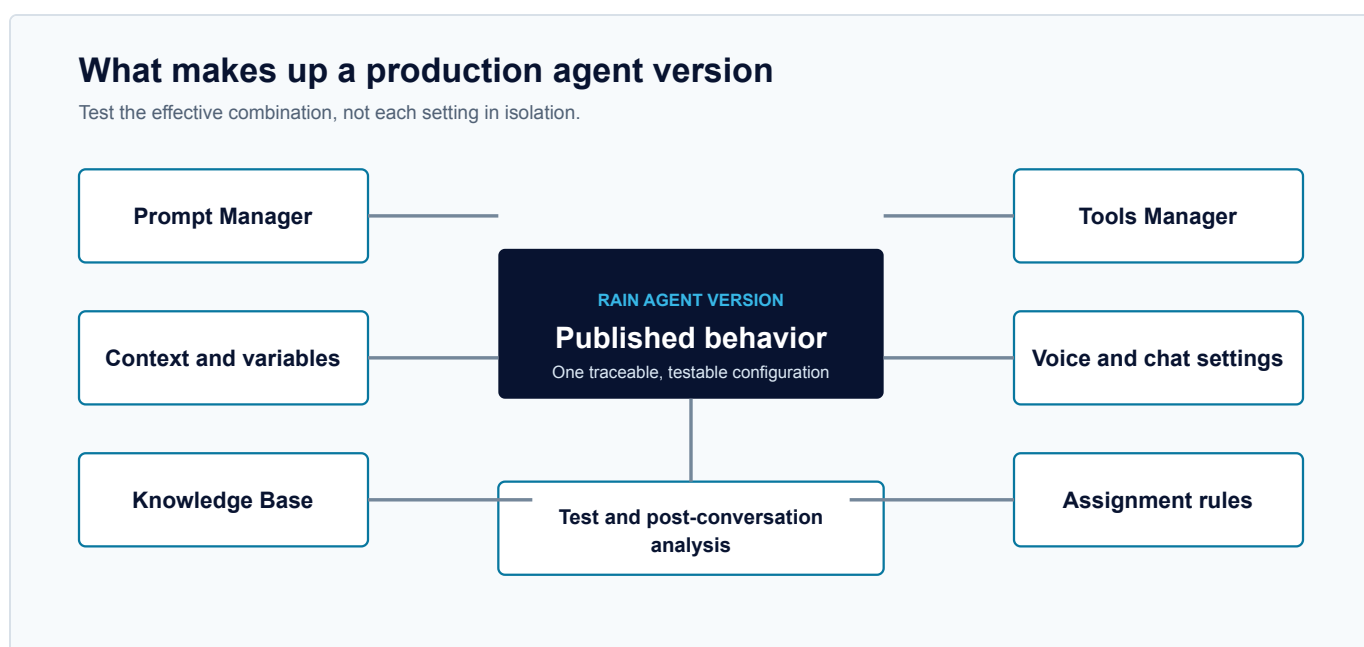
Troubleshooting

For a blank value, check object selection, field mapping, field access, record sharing, source value, relationship, and default. For the wrong value, verify that the selected record and object are correct and that one variable is not being populated by competing mappings. For excessive context, remove unused objects, record detail, activity history, variables, or knowledge rather than relying on the prompt to ignore them.

Test Environment and release validation

Use the browser Test Environment to validate a draft voice or Sales Coach experience, including record context, variables, live transcript, and tool results, before production channel testing.

Understand the test boundary



The effective prompt, context, tools, channels, and assignments must be tested together before production release.

The Test Environment starts a browser-based voice web call. For a Voice Agent, you speak with the agent. For Sales Coach, the AI plays the customer and you play the human seller. It is not a general chat test, and the browser test channel does not validate call transfer.

Use an editable draft version. A successful browser session proves only the web-call configuration and supplied test context. It does not prove phone routing, caller identity, quiet hours, voicemail, transfer, messaging, or production record access.

Before starting, confirm that the draft uses a supported voice or Sales Coach configuration, the required voice-provider account is connected and available, and the browser has working

microphone and audio access. Use the real phone or messaging channel later for capabilities the browser test does not exercise.

Prepare representative context

1. Select a Salesforce record when the version has an object source.
2. Review record-populated variables and enter approved manual overrides where needed.
3. Set inbound or outbound direction and confirm the configured time zone.
4. Open **Live Preview** when a record is selected and compare the effective values with the source record.
5. For Sales Coach, confirm the methodology, offering, customer profile, and role-play direction.

Manual overrides are test inputs. They do not prove the production object mapping. Use non-production or explicitly approved data and never enter credentials as variables.

Test behavior end to end

During the live call, review the transcript and any tool invocation and result entries. Include:

- Normal conversation and expected completion.
- Missing, contradictory, and restricted record context.
- Each supported tool's success, no-result, validation, and failure path.
- Confirmation and approval boundaries.
- Silence, interruption, and end behavior available in the web call.
- Transcript speaker labels and tool inputs and outputs.
- Prompt-injection and requests outside the authorized role.

Review the final transcript before resetting the session. A natural conversation is not sufficient when the wrong context or tool result was used.

Complete release validation elsewhere

After the web call passes, use Agents, versions, and publishing to publish the exact tested draft. Then run a controlled test through each real production channel and mapping. Compare the active version, runtime record, variables, permissions, provider state, transcript, durable Salesforce records, and follow-up work.

Salesforce recommends validating release behavior in a sandbox; sandbox preview guidance  explains the platform testing role. Provider-backed calls can still incur usage or contact external services in a sandbox, so confirm the connected account and test numbers first.

If test and production differ

Compare the published version, phone or chat mapping, runtime record, variables, permissions, provider synchronization, direction, environment, and enabled tools. Preserve the production Communication and transcript before retrying. Roll back the version or remove the mapping or assignment when behavior is unsafe.

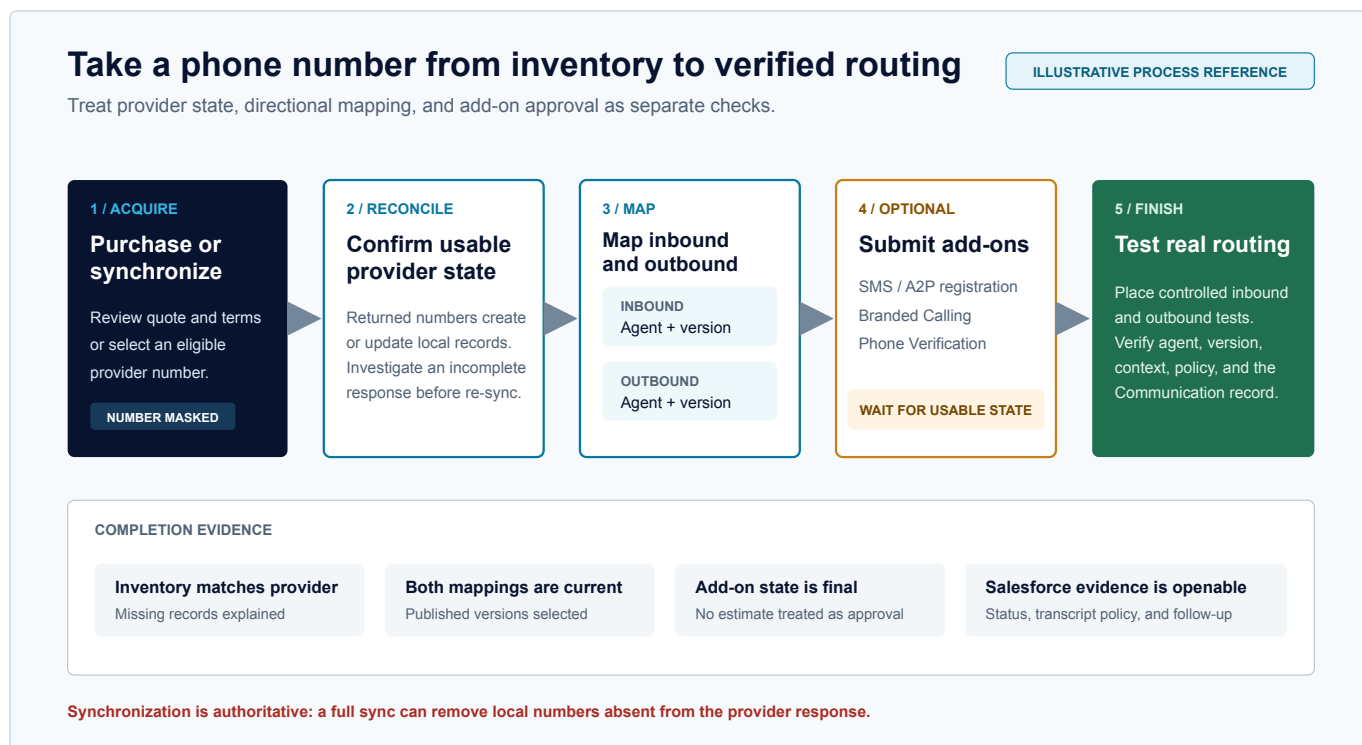
Phone numbers and add-ons

Purchase or synchronize phone numbers, map inbound and outbound agent versions, submit eligible add-ons, and reconcile Salesforce with provider status.

Add and map a number

Use the Setup Wizard or Phone Numbers tab to purchase a number or synchronize an eligible existing provider number. The current purchase flow supports available US and Canadian numbers from the configured providers. Search by area code, review the current checkout quote and terms, then wait for the Phone Number record to reach a usable state.

Configure separate inbound and outbound agent and version mappings when required. A number can route each direction differently, so verify both mappings after publishing a new version.



A number is ready only after provider reconciliation, inbound and outbound mapping, optional add-on checks, and an end-to-end routing test.

Configure eligible add-ons

The current add-on flow can manage:

- **SMS/A2P registration**, which requires business and campaign information before carrier review.
- **Branded Calling**, which requires a business profile and an authorization process before provider submission.
- **Phone Verification**, which requires business identity and attestation before provider review.

Add-ons move through provider-controlled states such as pending, submitted, in review, approved, active or verified, rejected, failed, deleting, and deleted. The exact path varies by add-on. An estimate shown in the interface is not a guaranteed approval date. Do not use an add-on until its final usable state appears on the Phone Number record.

Synchronize carefully

Provider synchronization is authoritative. A full synchronization creates or updates returned numbers and can delete local Phone Number records that are absent from the provider response. It also reconciles mappings, webhooks, and add-on states.

Confirm the connected provider account and preserve the current number and mapping inventory before synchronization during an outage, credential change, or provider migration. If the provider response looks incomplete, stop and investigate before repeatedly synchronizing.

Configure conversation behavior separately in Voice, call, and chat settings. Then test each mapped direction with the actual number. Confirm the correct agent and version, caller identity, context variables, end state, Communication record, transcript, recording policy, extracted values, and follow-up.

Finish and verify

Treat the number as ready only after the Salesforce inventory matches the intended provider inventory, both directional mappings point to published versions, every required add-on has reached a usable provider state, and controlled inbound and outbound tests produce the expected Communication records. Record an operating owner for provider reconciliation and future mapping changes.

Troubleshooting

Symptom	Check
Number is missing	Provider account, purchase or sync result, subscription, permissions, and refresh
Wrong agent answers	Inbound versus outbound mapping, active version, provider sync, and cached configuration
Number disappears after sync	Connected provider account, provider response completeness, and synchronization log
Add-on remains pending	Provider review status, submitted time, eligibility, required business information, and callback reconciliation
Add-on is rejected	Provider reason, business profile, authorization or attestation, campaign data, and resubmission guidance

Outreach Lists and next-best actions

Define an eligible audience, preview a voice or next-best-action use case, and control execution with channel, policy, quiet-hour, cooldown, redial, and concurrency settings.

Build an Outreach List

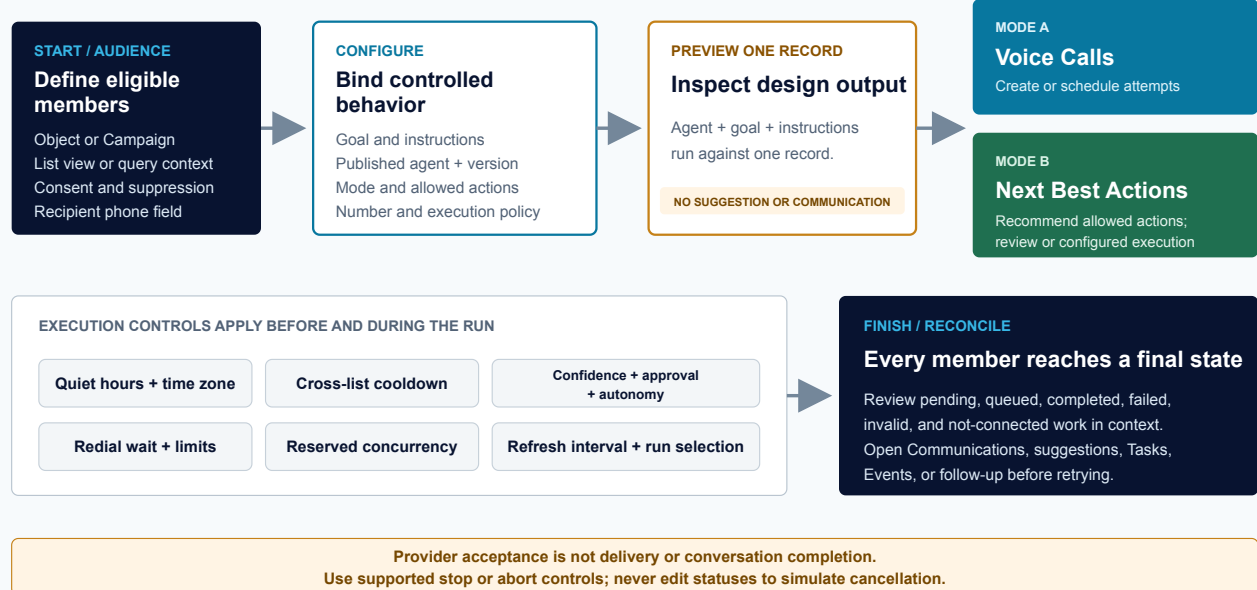
An Outreach List connects an audience definition with a published agent, mode, policy, and execution controls. Configure:

- Object and member type, including Lead or Contact membership when Campaign is the source.
- Source list view, Campaign, or other supported query context shown by the builder.
- Outreach goal and additional instructions.
- Mode, allowed actions, and execution policy.
- Agent and published version.
- From number and the recipient phone field.
- Quiet hours and time zone.
- Cross-list cooldown, redial wait, immediate and maximum redials.
- Reserved concurrency, refresh interval, and run selection.

Move an Outreach List from audience to final evidence

Preview one record, apply policy gates, then reconcile every member and resulting record.

ILLUSTRATIVE PROCESS REFERENCE



Preview one record before launch, apply timing and concurrency controls during execution, and reconcile every final voice or next-best-action outcome.

Preview the audience and exclusions before execution. A list view is not automatically a compliant audience; verify consent, suppression, ownership, duplicate records, prior contact, quiet hours, and the use case.

Voice Calls require an available voice and telephony provider, an approved number, and a published compatible agent. Next Best Actions require an available AI provider for the selected agent; SMS, Email, and AI Voice Call execution also require the corresponding provider-backed channel and approved sender. Task, Event, and Call Task actions depend on Salesforce access and the selected execution policy.

Choose voice calls or NBAs

Voice Calls creates or schedules communication attempts through the selected agent and number. **Next Best Actions** asks the selected agent to recommend from the allowed actions. The preview runs the configured agent, goal, and instructions against one record without creating a suggestion or communication; it is a design check, not a production result.

NBA action choices include SMS, Email, Task, Event, AI Voice Call, and Call Task. The interface presents **Recommendations Only** for review-oriented output and **Auto-Execute (Agentic)** to use

organization defaults. Confidence, approval, autonomy, permissions, channel readiness, and phone routing still govern execution; a batch request does not bypass them.

Use the Outreach List record and batch viewer to follow total, pending, queued, completed, failed, invalid, and not-connected work. Stop or abort through supported controls; do not edit status fields to simulate cancellation.

Completion and safe retries

Outreach is complete when every member has a final supported status and the resulting Communications, suggestions, Tasks, or follow-up can be opened. Provider acceptance alone is not delivery or conversation completion. Before retrying a failed or not-connected member, review attempt history, cooldown, redial policy, reply state, and prior results across lists.

Finish by reconciling the Outreach List totals to openable result records, assigning an owner to exceptions, and documenting whether a retry is allowed. Keep the list stopped when the audience, provider response, or result counts cannot be explained.

Troubleshooting

For an empty audience, check the selected object or Campaign, member type, list view, filters, access, phone field, and eligibility. For locked outreach, verify a published agent and linked number. For missing NBAs, review execution policy, allowed actions, confidence and approval rules, record context, phone routing for phone actions, and processing status. See Agent assignment rules for record-context prompt selection; those rules are a separate job from configuring an Outreach List.

Voice, call, and chat settings

Configure version-level voice, speech, call, voicemail, screening, and chat inactivity behavior, then validate each enabled channel independently.

Configure the opening and model behavior

The screenshot shows the RAIN OS interface for creating a new agent. The top navigation bar includes 'RAIN OS', 'RAIN OS Setup Wizard', 'RAIN Agents' (selected), 'Phone Numbers', 'Knowledge Base', 'Outreach Lists', 'Comms', 'Leads', 'Opportunities', 'Calendar', 'RAIN Hub', and 'More'. A search bar is also present. The main content area is titled 'New RAIN Agent' and asks 'What type of agent do you want to create?'. Below this, there are four options, each with an icon and a description:

- Assistant Agent**: Create an AI agent that understands your business, makes decisions, and puts you on auto-pilot.
- Voice Agent** (selected): Create an AI agent for phone calls and voice interactions.
- Chat Agent**: Create an AI agent for text-based chat conversations.
- Voice + Chat Agent**: Create an AI agent that handles both voice calls and chat.

A 'Next' button is located at the bottom of the options.

The selected agent type determines whether voice settings, chat inactivity controls, or both must be configured and tested.

In Agent Builder, choose who speaks first. When the AI speaks first, configure the welcome behavior and a pause from 0 to 5 seconds. When the user speaks first, the agent can begin after 0 to 20 seconds of silence. Review language, voice, model, and model temperature from 0 to 1; higher-priority model choices can have different usage cost.

Do not let a welcome message claim identity, consent, record facts, or success that has not been established. Test inbound and outbound openings separately.

Configure voice and speech

Voice speed ranges from 0.5 to 2. Voice temperature and volume range from 0 to 2. Speech controls include responsiveness, interruption sensitivity, backchannel behavior, and speech normalization. Background sound can be None, Coffee Shop, Convention Hall, Summer Outdoor, Mountain Outdoor, Static Noise, or Call Center, with volume from 0 to 1.

Use restrained settings first. High interruption sensitivity, aggressive responsiveness, or loud background sound can make a technically successful call difficult to use. Obtain compliance approval before adding sound that could misrepresent the caller's environment.

Configure call safeguards

- Voicemail detection can hang up or leave the configured prompt or static message.
- Call screening can identify the agent and purpose before the call is connected.
- End-on-silence ranges from 10 to 1,800 seconds.
- Maximum call duration ranges from 1 to 120 minutes.
- Ring duration ranges from 5 to 90 seconds.

Keep screening identity within the current 100-character limit and purpose within 300 characters. Treat the active UI as authoritative if it reports a changed range. Transfer configuration is separate from these safeguards and must be tested through a real phone call because the browser Test Environment does not support transfer.

Configure chat inactivity

Chat settings currently provide automatic closure for inactive chats. When enabled, the timeout ranges from 0.1 to 72 hours. Choose a duration that fits the support or sales workflow, preserve any required transcript, and explain how a customer can resume or start a new conversation.

Save, publish, and verify

Saving channel settings updates the Salesforce draft and can synchronize provider configuration. Confirm that the save and provider update both succeed. Test the exact draft, publish it, verify the phone or chat mapping points to that version, and repeat a controlled production-channel test.

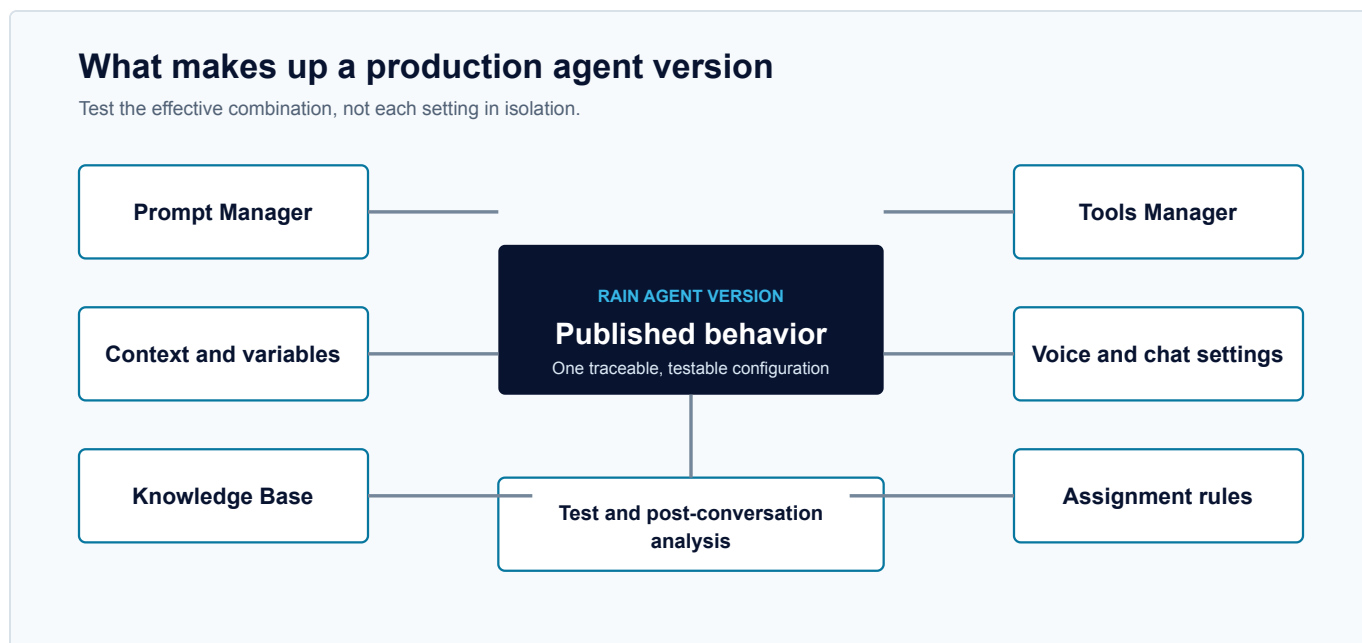
When behavior differs, compare the mapped version, provider synchronization, direction, number or chat entry point, browser versus real-channel limitations, and runtime permissions before

changing numeric settings repeatedly.

Agent assignment rules

Select a published agent prompt for a Salesforce record by configuring active object rules, explicit conditions, validated formulas, and deterministic priority.

Understand what a rule selects



Assignment rules connect a published agent version to record context; priority and eligibility determine which prompt runs.

An Agent Assignment Rule selects an agent prompt for a supported Salesforce record context. It does not assign record ownership and is separate from standard Salesforce lead, case, or territory assignment.

At runtime, active rules for the record's object are evaluated in ascending numeric priority. The first matching rule whose agent has a usable latest published version and Assistant prompt wins. A lower number runs first. A blank condition is unconditional, so place a catch-all rule last.

Build the rule

1. Select the target Salesforce object and agent.
2. Set a deliberate priority and leave the rule inactive while drafting.
3. Use the condition builder for straightforward field comparisons.
4. Set condition logic when several rows must be combined.
5. Use Advanced Formula only when the condition builder cannot express the requirement.
6. Validate the expression, save, and test representative records.
7. Confirm the selected agent has a latest published version with an Assistant prompt, then activate the rule.

The editor exposes fields available to the configuring user and supports relationship traversal where offered. Salesforce's formula operator and function reference [🔗](#) is useful for general syntax concepts, but RAIN's validator and runtime support only the functions accepted by the assignment editor. A formula accepted elsewhere in Salesforce is not automatically supported here.

Make priority deterministic

Keep overlapping rules rare and document why each higher-priority rule is narrower than the next. Test one record that should match each rule, one that matches several rules, one that reaches the catch-all, and one that should match none. Record the expected winning rule and agent version.

Changing a rule does not publish an agent draft. Changing an agent draft does not make it eligible until the required version and prompt are published.

Respect access boundaries

Assignment chooses a candidate prompt; it does not grant object, field, record, tool, channel, or provider access. Field availability in the editor and runtime access can differ by identity. Minimize fields used in conditions, avoid sensitive values where a less sensitive criterion works, and test with a representative non-administrator user. See Salesforce field-level security [🔗](#) for the platform boundary.

Troubleshooting

For no assignment, check object name, active state, condition values, formula validation, published version, and Assistant prompt. For the wrong agent, inspect all lower-numbered rules before changing the intended rule. For a rule that works only as an administrator, compare field access and

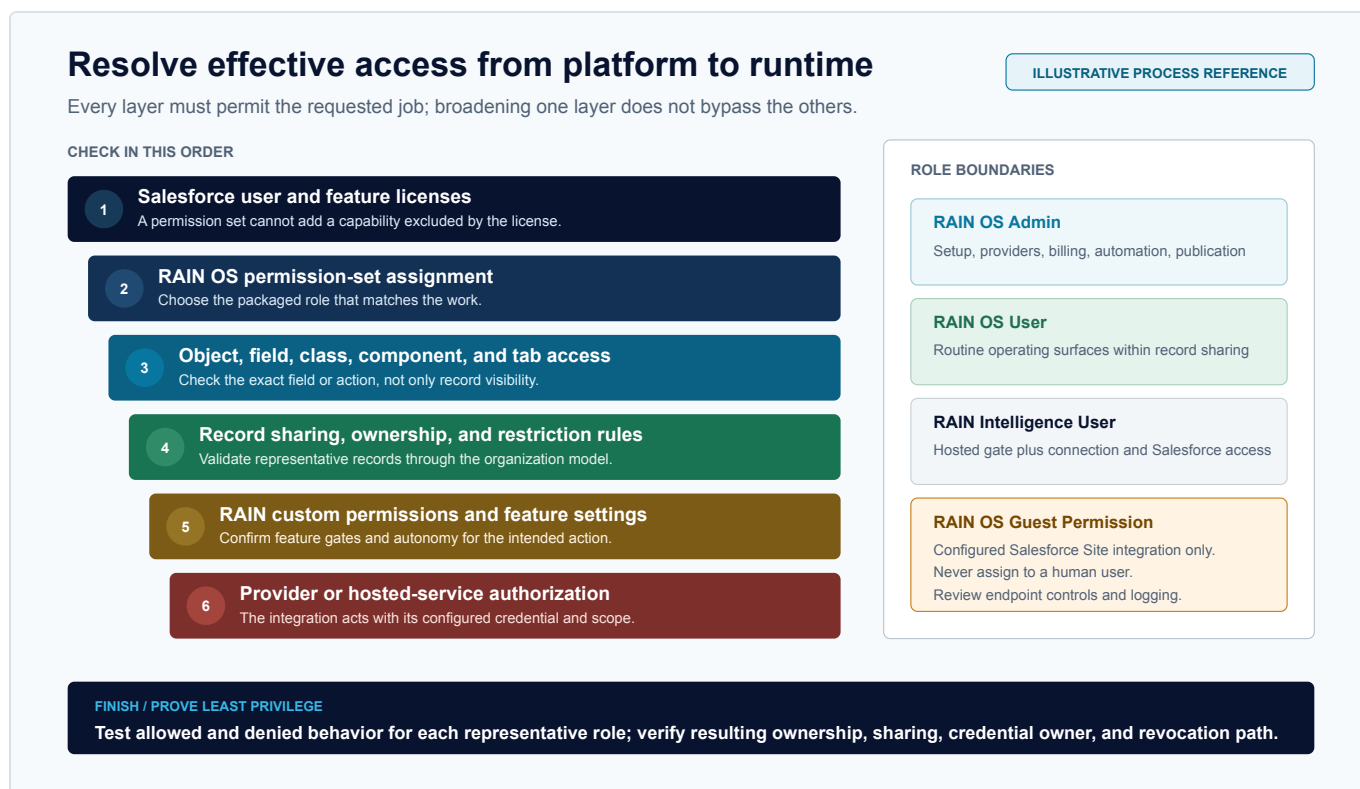
record sharing. For a changed result after publishing, confirm which version is now the latest published version for the selected agent.

Administration, permissions, and security

Apply layered Salesforce access, protect integration configuration, and grant each RAIN OS role only the capabilities it needs.

Access is layered

RAIN OS access is the intersection of several controls. A permission set can grant object, field, class, tab, or custom-permission access, but it does not override a missing Salesforce license, record sharing, restriction rules, or the access of an external integration credential.



Access depends on every layer in sequence, from licensing and permission sets through record sharing and the runtime identity used for the action.

Evaluate access in this order:

1. Salesforce user and feature licenses.
2. RAIN OS permission-set assignment.

3. Object and field permissions.
4. Organization-wide defaults, roles, teams, queues, and sharing.
5. RAIN custom permissions and feature settings.
6. Provider or hosted-service authorization.

Use Salesforce permission-set assignment  and User Access Summary  to verify platform access. Permission sets are additive and cannot grant a capability excluded by the user's license.

Packaged roles

RAIN OS Admin grants broad application, setup, API, reporting, metadata, and operational access, including administrative custom permissions. It is appropriate only for the small group that configures agents, providers, billing, automation, and package behavior.

RAIN OS User supports routine use of agents, communications, outreach, suggestions, insights, and related UI. Review its broad record access against the organization's sharing model before assigning it at scale.

RAIN Intelligence User is a narrow custom-permission gate for the optional hosted interface. Users also need an active hosted connection and appropriate Salesforce access.

RAIN OS Guest Permission is intended only for a configured Salesforce Site integration path. Guest execution has a different trust profile from authenticated user execution. Never assign it to a human user. Enable it only after reviewing the exact Apex and field grants, endpoint authentication, replay protection, and logging. Follow Salesforce's guest access security guidance .

Administrative responsibilities

RAIN OS administrators manage:

- Feature toggles and autonomy choices.
- Agent definitions, versions, assignments, and publication.
- Provider connections, sender identities, phone numbers, and credit status.
- Scheduled work and asynchronous failures.
- RAIN Intelligence enablement and connection state.
- Report and dashboard visibility.

Separate these duties where practical. In particular, a user who reviews suggestions does not normally need provider credentials, package setup, or billing control.

Configuration security

RAIN OS uses package-managed settings for feature behavior and integration configuration. Some settings are protected from subscriber code, but protected storage does not replace credential lifecycle management.

Administrators should:

- Enter credentials only through the supported setup surface.
- Avoid placing secrets in Flow variables, formulas, debug messages, reportable fields, or support screenshots.
- Rotate credentials when an administrator leaves, a provider reports exposure, or the integration scope changes.
- Use a dedicated least-privilege integration identity where the provider supports one.
- Revalidate webhooks and callouts after rotating a secret or changing a domain.

Do not infer that a connected provider has the same access as the user clicking a button. The provider acts with the credential configured for that integration.

Automation and autonomy

Feature settings can allow eligible recommendations or actions to execute with less review. Configure autonomy by business impact, not convenience. Start with explicit review, validate the generated inputs and resulting records, then relax only the action types and confidence bands that have stable outcomes.

Salesforce permissions remain one control, but system-context automation can have broader execution rights than an interactive user. Salesforce's Apex security guidance [🔗](#) explains why sharing, CRUD, and field security are distinct. Test with representative non-administrator users and inspect the resulting record ownership and sharing.

Access review

Review RAIN OS access at installation, quarterly, and whenever roles or integrations change:

1. Export current permission-set assignments.
2. Confirm each administrator still needs setup, billing, and provider control.
3. Confirm ordinary users can access only the records required by their role.
4. Verify guest access is still required and constrained.
5. Review active scheduled work and autonomous action settings.
6. Validate provider and hosted-service credentials and owners.
7. Test access-denied behavior for one user in each role.
8. Reconfirm who owns each provider, hosted-service, and webhook credential and how it is revoked.

Close the review only after each representative role can complete its intended work, is denied an out-of-role action, and produces records with the expected ownership and sharing. Record the reviewer, exceptions, credential owners, and next review date.

Troubleshooting access

When a user sees a missing tab, first check app navigation and permission assignment. When a record is visible but a field or action is not, check field access, component visibility, custom permissions, and record state. When a request starts but the integration fails, check the configured credential and provider authorization. When only some records fail, check sharing, ownership, and record-specific validation rather than broadening the permission set immediately.

Agents, versions, and publishing

Configure stable agent definitions, revise behavior through versions, and publish only after tools, knowledge, channel settings, and permissions are validated.

Agent and version roles

An agent is the durable Salesforce definition for a business role. An agent version is a revision of its instructions, model or channel behavior, tools, knowledge, and provider configuration. Keeping those concepts separate lets administrators prepare a change without silently altering the active experience.

Assignments determine where a published prompt applies. Phone and chat configuration determine which version receives channel traffic. A published version is intended for live use; a draft remains editable and should not be treated as active.

Prerequisites

Before creating an agent:

- Define the user job, audience, permitted data, and success criteria.
- Confirm the channel and provider are configured.
- Identify required Salesforce tools, Flows, and knowledge.
- Decide who may edit, review, publish, and operate the agent.
- Select a test record set that contains realistic success, empty, access-denied, and validation-failure cases.

Create and revise an agent

The screenshot shows the 'New RAIN Agent' configuration page in the RAIN OS interface. The top navigation bar includes 'RAIN OS', 'RAIN OS Setup Wizard', 'RAIN Agents' (selected), 'Phone Numbers', 'Knowledge Base', 'Outreach Lists', 'Comms', 'Leads', 'Opportunities', 'Calendar', 'RAIN Hub', and 'More'. A search bar is also present. The main content area asks 'What type of agent do you want to create?' and 'Select the channel your agent will operate on.' Below this are four cards: 'Assistant Agent' (AI agent for business decisions), 'Voice Agent' (AI agent for phone calls and voice interactions, marked with a checkmark), 'Chat Agent' (AI agent for text-based chat), and 'Voice + Chat Agent' (AI agent for both voice and chat). A 'Next' button is at the bottom of the cards. The bottom status bar shows 'RAIN (6)', 'Phone', 'To Do List', and 'Notes'.

Agent type is chosen before the agent identity and first draft, so channel scope is established before versioned configuration begins.

1. Create the agent definition with a clear business purpose and owner.
2. Create or clone a draft version.
3. Configure instructions, channel options, tools, variables, and knowledge.
4. Add assignment rules or channel mappings only after the draft has testable behavior.
5. Validate the draft against representative records and users.
6. Review provider-specific settings and expected data transfer.
7. Publish or synchronize the version.
8. Confirm the provider and Salesforce show a consistent active state.

Use a new version for material behavior changes. Editing a draft is appropriate while testing; changing active behavior without a reviewable version weakens rollback and auditability.

Publishing and synchronization

Publishing can require an external provider call. Salesforce records the local lifecycle, while the provider can return its own identifier and status. A local save does not guarantee that remote synchronization completed. Publish rights should remain narrower than edit or routine-use rights.

After publishing, verify:

- The intended version is active.
- Required tools and knowledge are attached.
- The provider reports a usable agent where a provider is required.
- The correct phone number, chat entry point, or assignment uses the version.
- A low-risk test interaction produces the expected Salesforce activity.

If knowledge or a tool changes, the affected provider-backed version may require synchronization again.

Release first to a controlled number, chat entry point, assignment, user, or audience. Keep the prior published version available as a rollback reference, and record which mappings must change during rollback.

Clone and convert operations

Cloning creates a new starting point while preserving the original agent. Conversion actions can prepare an agent for another supported channel, but channel-specific settings still require review. Voice needs telephony, pronunciation, transfer, and post-call choices that chat does not. Chat needs entry-point and response behavior that may not apply to calls.

Treat conversion as configuration assistance, not proof of channel readiness.

Permissions and ownership

Agent administrators need access to the agent, version, tool, assignment, knowledge, and provider setup records involved. Operators may need only the published experience and related business records. Keep publish rights narrower than routine use rights.

Record sharing can also affect testing. An administrator's successful test does not establish that a standard user or runtime integration can read the same records.

Troubleshooting

A draft cannot publish: Check required fields, version status, tool validation, provider configuration, and access to linked records.

Salesforce says published but the channel uses old behavior: Check remote synchronization, the version selected by the assignment or phone number, and cached provider state. Avoid repeatedly publishing until the first request's status is understood.

Knowledge changes are absent: Confirm the knowledge link is active and the version has been synchronized after the change.

Only one user or record type fails: Test sharing, field access, assignment criteria, and required merge data with that context.

A conversion produces incomplete behavior: Review every channel-specific setting and republish only after a channel test.

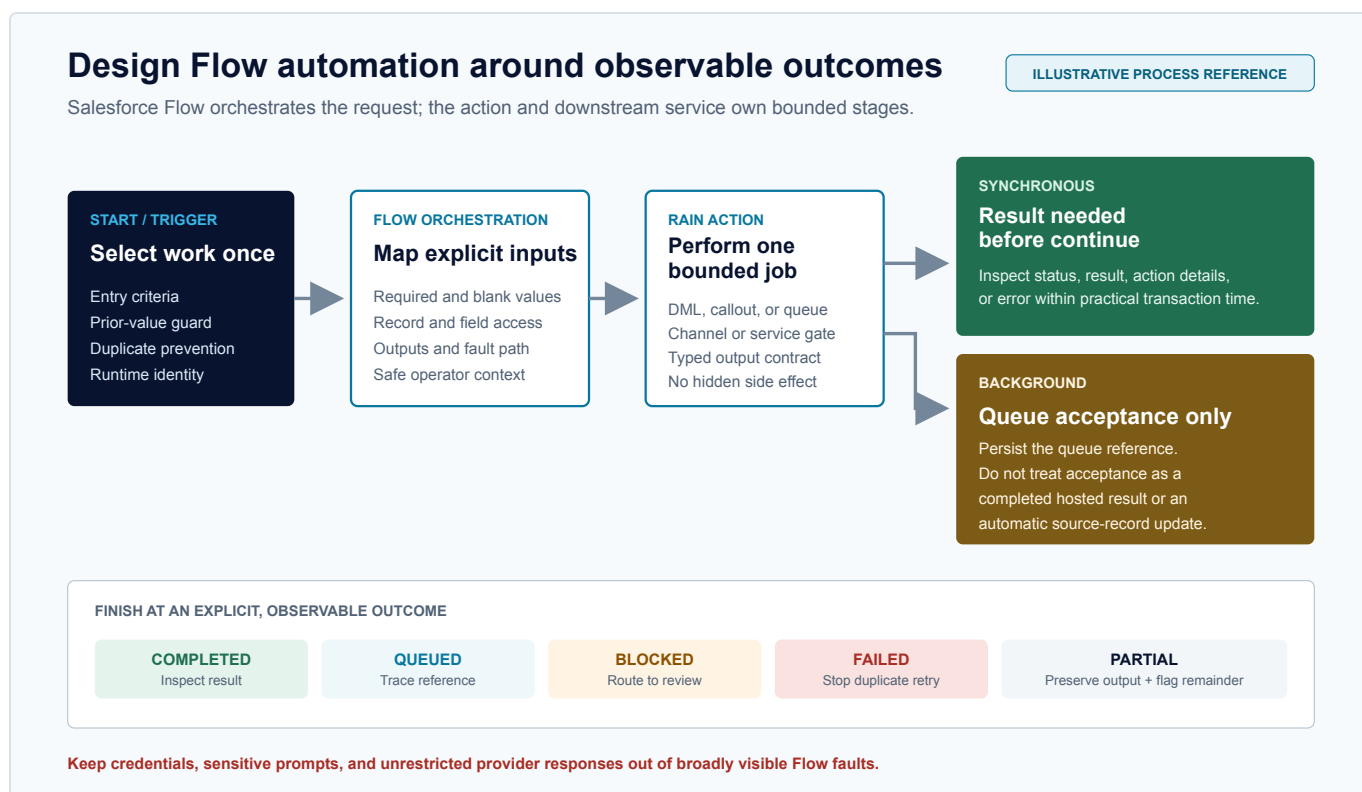
Automation and Flow actions

Use RAIN OS invocable actions and scheduled Flows with explicit inputs, least-privilege execution, observable outputs, and separate handling for Salesforce and hosted background work.

Automation surfaces

RAIN OS supplies active Salesforce Flows for suggestion lifecycle work, scheduled ownership updates, scheduled SMS, and scheduled voice calls. It also supplies invocable Apex actions for communication, voice, formatting, synchronization, and the optional RAIN Intelligence service.

The Flow owns orchestration in Salesforce. The invocable action owns its bounded operation. An external provider or RAIN Intelligence may own a later stage.



The Flow must prevent duplicate entry, supply explicit inputs, choose the correct execution branch, and leave every outcome visible to operators.

Design an automation

Before adding a RAIN action to a Flow:

- Identify the triggering event, entry criteria, and duplicate-prevention rule.
- Define required inputs and their behavior when blank.
- Determine whether the action performs DML, a callout, or queued work.
- Confirm the Flow user and runtime context can access the target records and fields.
- Map outputs and Flow fault paths  to an observable status or operator queue.
- Test interaction with existing Apex, validation rules, and other active Flows.

Avoid using an invocable action as an undocumented side effect. A future administrator should be able to tell what request was made and where to inspect its result.

Communication actions

The SMS and voice actions require a valid target, channel configuration, sender or phone context, and provider readiness. The action can create or queue communication work; final delivery or call completion remains asynchronous.

For scheduled work, store the requested time in a consistent Salesforce time-zone context and test daylight-saving transitions. Follow Salesforce's scheduled-path guidance  when configuring timing, and ensure the Flow selects each due record once.

Run RAIN Intelligence from Flow

The public **Run RAIN Intelligence** action accepts a goal plus optional record, object, agent-version, file, model, timeout, execution-mode, safety-gate, and destination choices. It returns success and status information, a run reference, result or summary data when available, action details, and an error when the request fails.

Use synchronous execution when the Flow needs the hosted result before continuing and the task can finish within the transaction's practical callout time. Use background execution when the Flow only needs to enqueue the request.

Background mode queues Salesforce work and returns queue acceptance. It does not return the completed hosted analysis to the Flow or automatically update the source record. Do not design a later Flow decision as though a queued response contains the completed result.

Safety-gate choices range from planning or asking before changes to broader autonomy. Use review-oriented modes for writes, deployments, customer-visible actions, and ambiguous record targets.

Finish with observable outputs

Handle these outcomes explicitly:

- **Completed:** inspect returned result and any action details.
- **Queued:** persist the queue reference if operators need to trace it; do not treat it as completion.
- **Blocked or awaiting input:** route to review rather than retrying automatically.
- **Failed:** store a safe error summary and the affected record, then stop duplicate retries.
- **Partial:** preserve useful output but flag that not every requested action completed.

Do not place credentials, full sensitive prompts, or unrestricted provider responses into Flow fault emails or broadly visible fields.

The automation is complete only when every branch ends in a status an operator can find and interpret. Verify one successful path, one blocked or fault path, and duplicate prevention with the same runtime identity and sharing model used in production.

Troubleshooting

The Flow cannot find an action: Check package version, action access, Flow type, and whether the user can run Apex.

Inputs are rejected: Compare required values and data types, then check null handling and record context.

A synchronous hosted call times out: Narrow the goal or use background mode only when completion is not needed in the same Flow.

A background request never updates the record: Background mode does not update the source record with a completed result. Add an explicit supported persistence step or use synchronous execution for that requirement.

The Flow runs twice: Review trigger criteria, prior-value checks, recursion, scheduled selection, and downstream record updates before adding another guard.

Communications and providers

Configure senders and providers, create communication records, and use delivery status and reconciliation rather than submission alone to judge outcomes.

Communication model

RAIN OS uses a common Salesforce communication record for supported SMS, voice, and chat activity. The record holds the channel, sender and recipient context, message or call details, lifecycle status, provider references, and failure information. Email composition uses Salesforce email behavior and its native activity records.

A communication record is the operational audit point. A composer confirmation means Salesforce accepted the request. Delivery succeeds only when the configured provider processes it and RAIN OS receives or reconciles the result.

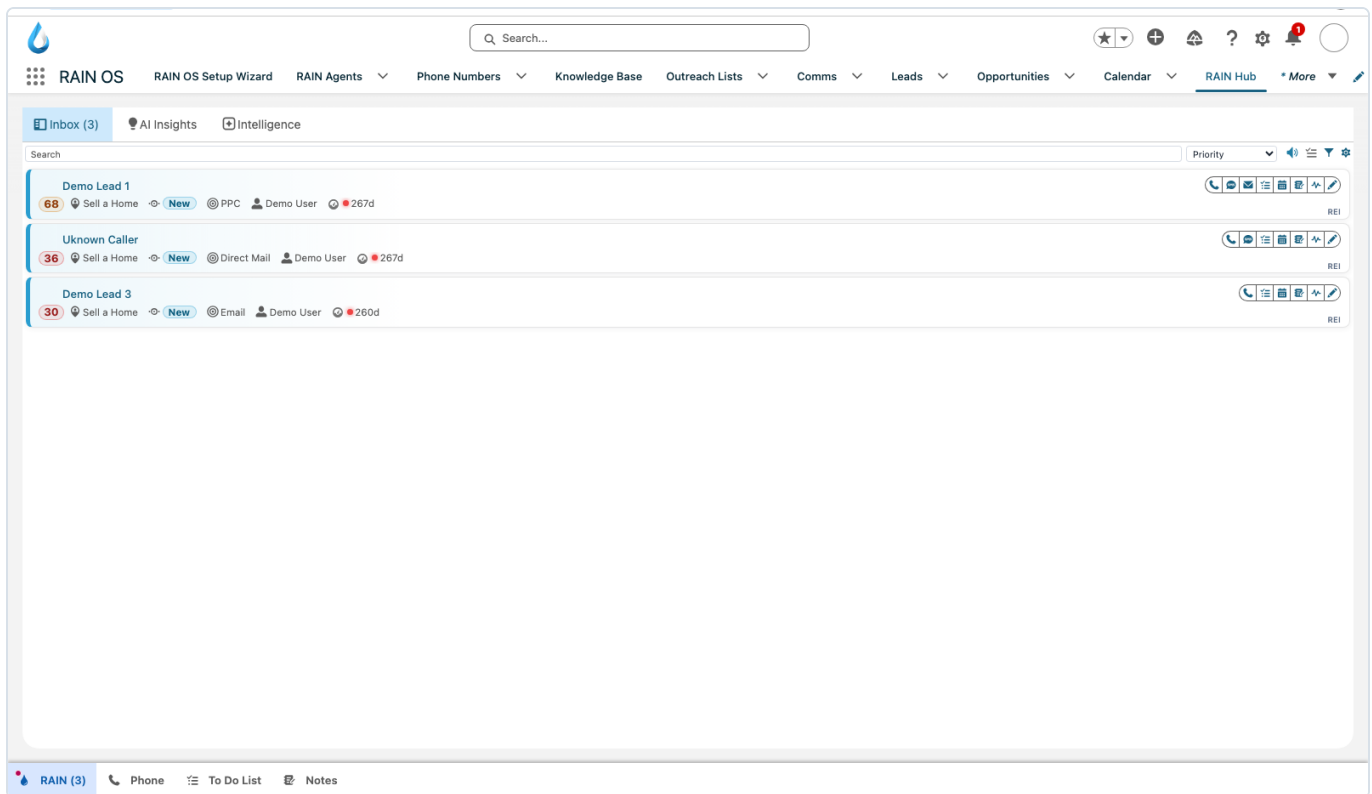
Prerequisites

Before enabling a channel:

- Configure and authorize a supported provider.
- Synchronize or create an approved sender identity.
- Confirm recipient fields and consent rules for the business process.
- Assign users access to the composer, communication records, and source records.
- Validate available credits, provider account status, sending limits, and any call-center configuration.
- Review templates and merge fields for missing-data behavior.

Provider availability varies by channel and organization configuration. A provider name in package capability does not mean it is enabled in every subscriber org.

Send a communication



RAIN Hub exposes record-level communication entry actions beside each synthetic work item. Delivery lifecycle, provider reconciliation, message content, and recipient details are intentionally not shown.

1. Open the communication composer from the relevant record or RAIN OS workspace.
2. Select the channel and an approved sender.
3. Resolve or enter a valid recipient.
4. Select a template or enter content, then review all merged values.
5. Send immediately or choose a supported scheduled time.
6. Confirm that a communication record was created.
7. Monitor its status until it is sent, delivered, failed, or otherwise reaches the provider's terminal state.

For bulk or automated outreach, validate one representative communication before increasing volume.

Templates and merge values

Templates improve consistency but can fail when a source record or merge value is missing. A preview is the correct place to catch an empty name, stale phone number, invalid link, or text that

exceeds provider limits.

Treat template edits as behavior changes. Re-test each channel that uses the template, especially when a merge value crosses object relationships or depends on optional data.

Scheduling and asynchronous delivery

Scheduled SMS is selected and dispatched by Salesforce automation when its due time arrives. Provider dispatch then happens asynchronously. The scheduled time, Salesforce job execution, provider acceptance, and final delivery are separate events.

Do not create a second message merely because the status has not advanced immediately. First determine whether the record is waiting for its scheduled time, queued for dispatch, accepted by the provider, or awaiting a delivery update.

Status and reconciliation

Use status, timestamps, provider references, and error details together:

- A pending or queued record has not reached final delivery.
- A sent record was accepted for sending but may not prove handset delivery.
- A delivered state reflects provider confirmation when that provider supplies it.
- A failed state should include enough category or detail to correct the input, credential, credit, or provider issue.

Inbound communications and provider callbacks can update an existing thread or create related activity. Reconciliation handles delayed provider results and should be allowed to finish before manual duplication.

Security and compliance

Limit who can select high-trust senders or view sensitive message content. Follow the organization's consent, opt-out, quiet-hours, recording, and retention policies. Salesforce sharing controls who can view the communication record; the provider independently stores or processes data under its own agreement.

Troubleshooting

No sender is available: Check provider connection, sender synchronization, channel support, user access, and sender status.

The recipient is rejected: Confirm format, required country information, source-field access, consent state, and provider restrictions.

The record remains pending: Check scheduled time, asynchronous jobs, dispatch eligibility, credits, and provider availability.

The provider accepted the request but status is stale: Check callback or reconciliation health before resending.

Merged content is wrong: Correct the source data or template mapping and create a new reviewed message; do not overwrite an already sent audit record.

Insights, suggestions, and scoring

Interpret RAIN insights and scores as decision support, and apply suggestions only after validating their evidence, target, permissions, and resulting record change.

Three related outputs

An **insight** explains a detected pattern, event, or piece of decision support. A **suggestion** proposes a reviewable change or next action. A **RAIN score** provides a prioritization signal for supported records.

These outputs can be related, but they do not have the same lifecycle. An insight can be useful without changing a record. A suggestion is not applied until its action succeeds or configured autonomy permits it. A score should be interpreted with its profile, refresh time, and the quality of available data.

How results are created

RAIN OS can generate analysis after relevant Salesforce activity, through asynchronous processing, by scheduled refresh, or from an enabled agent or provider workflow. The result is written to Salesforce for review and reporting.

Generation depends on accessible source data, configured models or providers, eligible record state, feature settings, and asynchronous capacity. The absence of a result may mean the record was ineligible or processing failed; it does not always mean that the analysis found nothing.

Review an insight

Review the subject record, source activity, creation time, summary, confidence or supporting context, and whether more recent data changes the conclusion. Use the insight to inform work; verify consequential facts in the authoritative Salesforce record or connected system.

Mark or manage the insight through the supported UI so view and disposition history remain available to reporting.

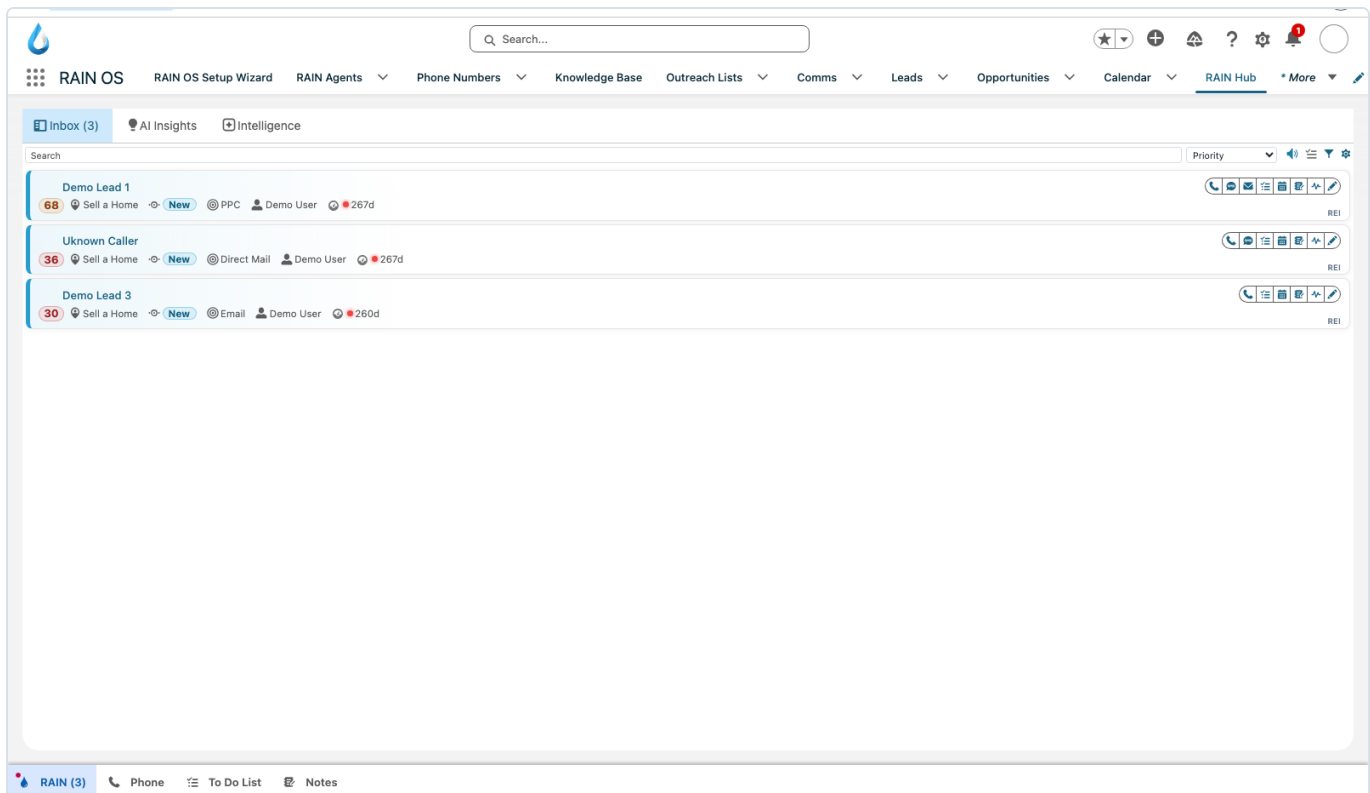
Review and apply a suggestion

1. Open the suggestion and its target record.
2. Confirm the proposed field change or next-best action is still relevant.
3. Check the evidence, confidence, current field value, ownership, and downstream automation.
4. Approve, reject, or defer using the supported status action.
5. Confirm the record change or action actually succeeded.
6. Review any follow-on communication, Flow, or provider status separately.

RAIN OS should not mark an unsuccessful field update as a successful approval. If a suggestion remains unapproved, inspect the application error rather than changing the status manually.

Autonomous settings can approve eligible suggestions based on action type and confidence. Keep sensitive fields and customer-visible actions under explicit review until production outcomes are stable.

Use RAIN scores



The Inbox keeps a bounded score beside the work item and exposes AI Insights as a separate review tab. The screenshot does not show evidence or an apply-suggestion decision.

RAIN scoring supports prioritization for configured record types. Administrators define or train the scoring profile, queue refresh work, and backfill eligible records. Users should interpret the score as one signal alongside stage, age, activity, owner context, and source-data completeness.

Scores can become stale when source data changes or scheduled refresh is delayed. Confirm the score's last update before using it for a large assignment or outreach decision.

Permissions and side effects

Users need access to the insight or suggestion and its target record. Applying a suggestion can trigger validation rules, Flows, Apex, assignment rules, or external integrations. Test the complete transaction with a standard user, not only an administrator.

Troubleshooting

No insight or suggestion appears: Check feature settings, eligibility, source activity, provider configuration, asynchronous jobs, and record access.

Approval fails: Check current field value, field access, validation, record lock, sharing, and downstream automation.

A duplicate proposal appears: Compare target, action, prior approved history, and generation timing before acting on either record.

A score is blank or stale: Confirm the record type is included, the scoring profile is active, training or backfill completed, and refresh automation is healthy.

Results look systematically wrong: Pause autonomous action, sample both positive and negative outcomes, inspect source-data completeness, and retrain or reconfigure only after identifying the bias.

Install, upgrade, and initialize RAIN OS

Install RAIN OS first, assign access deliberately, and verify each asynchronous initialization step before enabling production use.

Prerequisites

RAIN OS is the first package in the RAIN product installation order because it declares no package dependencies and owns the Salesforce data model used by its optional integrations.

Before installation:

- Verify the publisher, package version, target org, Salesforce edition, package requirements, org limits, and required API or Sites capabilities. Salesforce provides a before-install checklist 📄.
- Use an installer who can install managed packages, assign permission sets, schedule Apex, and configure any required site or integration access.
- Decide which users are administrators, standard RAIN OS users, hosted RAIN Intelligence users, and, only when required, site guest users.
- Inventory existing scheduled jobs, communication providers, call-center configuration, and automation that could overlap with RAIN OS.
- Establish a sandbox validation and rollback plan for upgrades.

Install into a representative sandbox first. Record the current package version, active agents, number mappings, credentials owners, schedules, and critical operational totals before an upgrade.

Installation lifecycle

Connect to Salesforce
Authorize RAIN Intelligence to access your Salesforce org

Environment

Production
login.salesforce.com

Sandbox
test.salesforce.com

RAIN Intelligence will have access to:

- ✓ Read and query your data
- ✓ Access metadata and schema
- ✓ Create, update, and deploy (when requested)

Want RAIN inside Salesforce?
Install the native RAIN OS package after connecting, or install it first if you are the Salesforce admin.

[Install RAIN OS Package](#)

[Connect with Salesforce →](#)

You can revoke access anytime from your Salesforce settings

For Intelligence-enabled deployments, package installation and hosted Salesforce connection are separate actions. The native Setup Wizard remains the authoritative post-install initialization surface and is not shown here.

Follow Salesforce's managed-package installation procedure , using the approved RAIN package link. The package install starts a post-install process. The synchronous portion establishes runtime access, assigns the installer administrative access, initializes critical defaults, and requests deferred setup. On a fresh install it also enables initial application navigation. An upgrade preserves existing configuration where possible and does not repeat every fresh-install presentation step.

Deferred initialization performs independent setup operations such as metadata synchronization, agent seeding, provider synchronization, credit defaults, site setup, scoring refresh, and scheduled work. These operations are deliberately isolated so one failed integration does not roll back every successful setup step.

That isolation also means package installation can finish while one or more capabilities remain uninitialized. Treat installation success as the start of validation, not proof that every feature is ready.

Post-install validation

After installation or upgrade, verify:

1. The installer has the intended administrator permission set and ordinary users have only the access they need.
2. The RAIN OS application and required tabs are visible to the intended users.
3. Agent, communication, insight, suggestion, outreach, and phone records can be opened according to role.
4. Expected scheduled or queued work exists once, with no obsolete duplicate schedule.
5. Default settings exist without replacing subscriber-managed values.
6. Provider connections, phone-number synchronization, and hosted-service connection status match the features being enabled.
7. The default agents needed by the organization exist and can be opened before publication.
8. Reports and the dashboard are accessible through the intended folders and running-user context.
9. The Setup Wizard recalculates required readiness from the expected terms, site, webhook, and feature settings.

Repeat the validation after a sandbox refresh, credential rotation, package upgrade, or material permission change.

Assign access

Use the packaged permission sets as role starting points:

- **RAIN OS Admin** is broad operational and setup access. Assign it only to trusted administrators.
- **RAIN OS User** supports ordinary application work but still depends on Salesforce licenses, object access, and record sharing.
- **RAIN Intelligence User** enables the optional hosted entry point; it does not configure or license the hosted service by itself.
- **RAIN OS Guest Permission** is for a specifically configured guest integration path. Do not assign it to human users or a site guest unless that endpoint is required and reviewed.

The installer receives administrative access automatically. Other administrators and users still require deliberate assignment.

Upgrade procedure

For an upgrade, review Salesforce's managed-package upgrade considerations 🌩, then:

1. Read the release scope and identify changed objects, automations, permissions, provider contracts, and reports.
2. Test the upgrade against representative configuration and data in a sandbox.
3. Record existing settings, active agent versions, provider connections, scheduled jobs, and critical report totals.
4. Install with a user who can complete post-install work.
5. Re-run the post-install validation and compare results with the recorded baseline.
6. Exercise one low-risk transaction in each enabled channel before resuming high-volume automation.

Do not delete or recreate settings merely because a new default was introduced. Initialization is intended to backfill missing defaults while retaining existing choices.

Troubleshooting initialization

If one area is unavailable while the application otherwise loads, treat it as partial initialization. Review Salesforce setup logs, scheduled jobs, asynchronous job failures, provider status, and the affected RAIN OS record. Retry only the failed setup or synchronization after correcting its prerequisite.

If a job appears more than once, pause the affected automation before removing a duplicate. Confirm which schedule is current and whether the old schedule is still producing work.

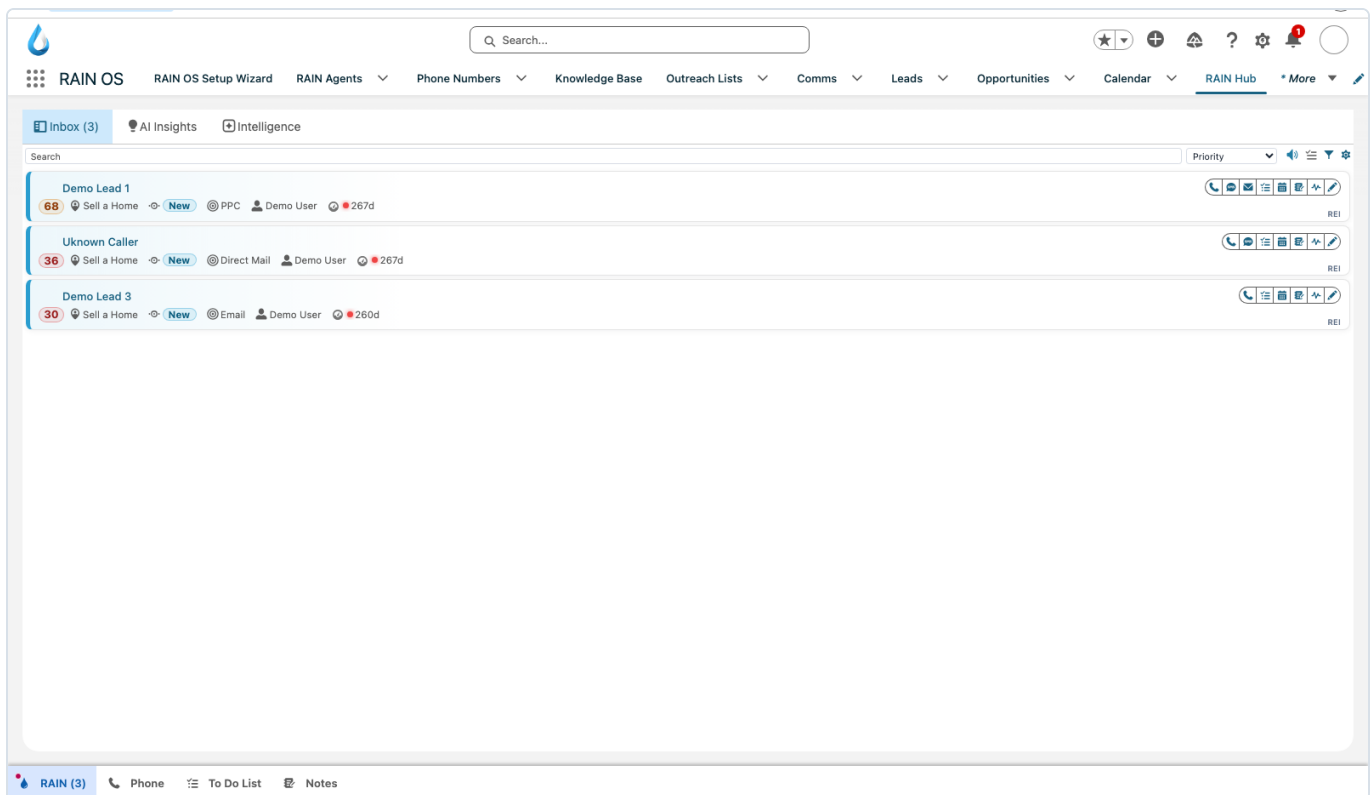
If tabs or components are missing, distinguish permission assignment from application navigation and Lightning-page activation. If a provider feature fails, confirm its connection separately from package installation. If RAIN Intelligence is unavailable, validate the hosted compatibility checklist without changing standalone RAIN OS configuration.

OPERATIONS

Operations and troubleshooting

Diagnose RAIN OS by following the record from Salesforce validation through asynchronous work and provider completion without duplicating an in-flight action.

Start with the affected record



The Inbox provides a safe starting record and visible operational context before deeper troubleshooting. Communication lifecycle detail, logs, retries, and provider evidence require their own surfaces.

Most RAIN OS processes leave a Salesforce record that identifies the request and its status. Start there instead of retrying from the UI. Record the channel or feature, current status, relevant timestamps, target record, user role, and whether a provider reference exists.

Classify the failure stage:

1. UI or permission prevented submission.

2. Salesforce validation rejected the input.
3. Apex or Flow failed before queueing work.
4. Asynchronous work failed or remains queued.
5. The provider rejected or delayed the request.
6. A provider callback or reconciliation step failed.
7. Post-processing failed after the primary action succeeded.

This classification determines whether a retry is useful and reduces duplicate communication or record changes.

Common checks

Access: Permission-set assignment, object and field access, record sharing, app navigation, component visibility, and integration credential scope.

Configuration: Feature enablement, provider connection, sender or phone status, active agent version, required templates and knowledge, and credit availability.

Data: Required fields, valid recipient, current record state, ownership, consent, merge values, and validation rules.

Automation: Flow entry criteria, duplicate guards, scheduled time, Apex jobs, fault-path handling , platform-event delivery, locks, and limits.

External service: Authorization, account state, provider status, request limits, callback processing, and remote identifiers.

Retry decisions

Retry only after identifying a transient stage and confirming the original request will not still complete. Good retry candidates include a temporary provider outage or a corrected expired credential. Bad retry candidates include invalid input, missing consent, a validation rule, insufficient access, or an ambiguous status with no reconciliation check.

For communications and calls, use the existing provider reference and terminal status to prevent duplicate customer contact. For suggestions, confirm the target record was not already changed. For agent publication, check whether the provider accepted the first synchronization.

Scheduled and asynchronous work

Use Salesforce's scheduled-job and asynchronous-job views to find failed, aborted, or long-running work. For recurring Apex work, compare the configuration with Salesforce's scheduled Apex guidance 🌩. Correlate job time with the affected RAIN OS record; a successful parent job can still enqueue child work that later fails.

If a schedule is duplicated, pause the feature before changing jobs. Identify the current schedule and inspect whether both jobs selected the same work.

Safe support evidence

Provide:

- Product and package version.
- Feature and channel.
- Approximate event time and time zone.
- Salesforce record type and a support-safe reference.
- User role and permission-set category.
- Current status and sanitized error category.
- Whether the issue is reproducible in a sandbox.
- Recent upgrade, credential, provider, or automation changes.

Do not provide credentials, tokens, full private endpoints, message content, recordings, transcripts, or unrestricted debug logs unless an approved secure support process explicitly requires them.

Recovery priorities

Protect customers and data first. Pause high-volume or autonomous work if duplicates, incorrect targeting, permission expansion, or unreviewed writes are possible. Preserve records and provider references for investigation. Restore read-only and review workflows before resuming autonomous or bulk execution.

If only RAIN Intelligence is affected, isolate hosted-dependent entry points and keep verified standalone RAIN OS functions available.

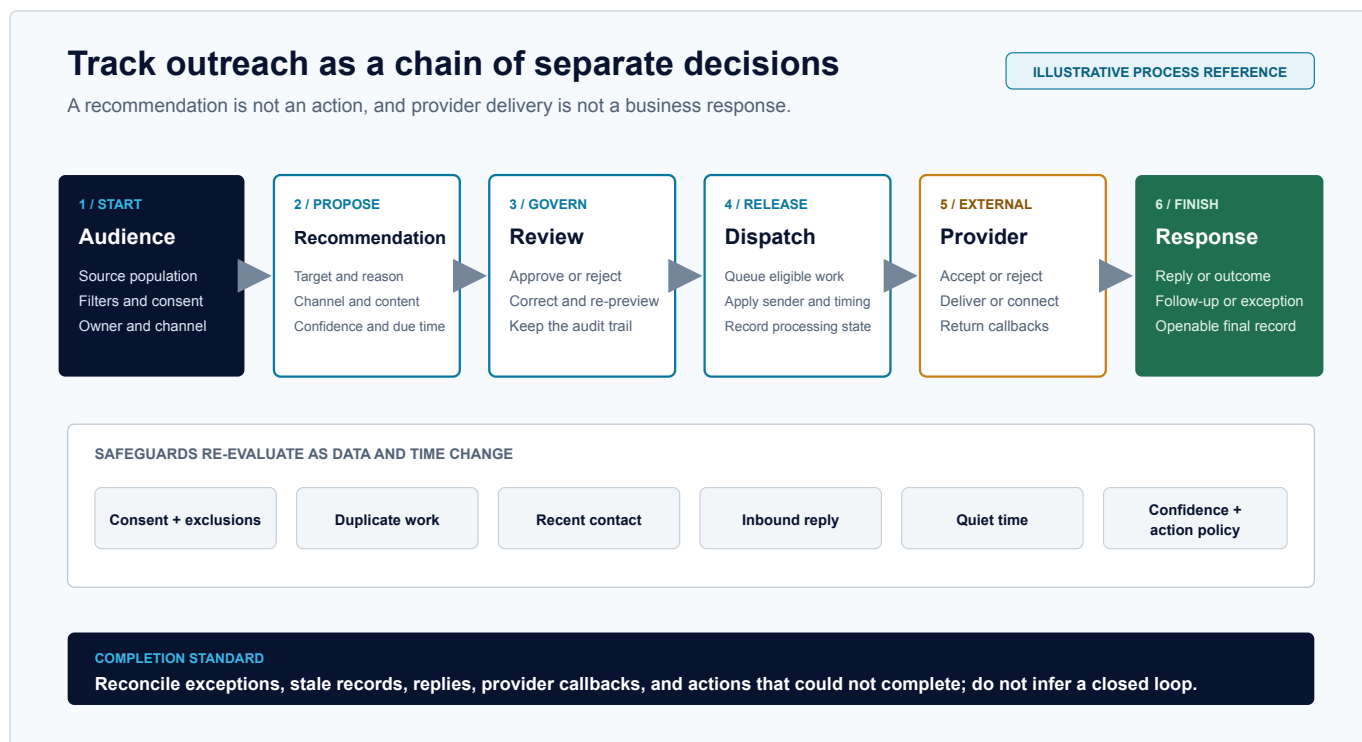
Outreach and next-best actions

Build an eligible outreach audience, preview recommended actions, and control execution with timing, duplication, reply, confidence, and approval safeguards.

Outreach model

An outreach list defines a Salesforce audience and a repeatable communication or action process. RAIN OS can generate list members or communication work, preview next-best actions, and route eligible actions for human review or configured autonomous execution.

The recommendation is not the action itself. Audience generation, recommendation, approval, communication dispatch, provider delivery, and business response each have their own status.



Recommendation and human review remain separate from dispatch, while consent, suppression, timing, and reconciliation safeguards span the full lifecycle.

Prerequisites

- Define the campaign purpose, audience criteria, channel, sender, and owner.
- Confirm source records contain usable contact data and required consent.
- Configure an available AI provider for the selected agent before previewing or creating next-best actions.
- Configure the communication provider, approved sender, and templates before provider-backed dispatch.
- Establish duplicate, recent-contact, inbound-reply, quiet-time, and exclusion policies.
- Decide which action types always require review.
- Validate confidence and timing settings in a sandbox or limited audience.

Build and preview an outreach list

1. Create the outreach list and select the intended source population.
2. Configure filters, channel, template, sender, and schedule.
3. Generate or refresh the list.
4. Review included and excluded records, especially boundary cases.
5. Preview next-best actions and merged content.
6. Correct source data or criteria before approving work.
7. Release a small batch and monitor through delivery and response.

Refreshing a list can change eligibility as source data, prior communication, replies, or timing changes. Record the business meaning of the filters instead of assuming a previous audience remains valid.

Recommendation safeguards

RAIN OS can suppress or defer outreach when it detects recent contact, duplicate work, an inbound reply, an ineligible schedule, or a disabled feature. Confidence thresholds and per-action settings can determine whether an eligible recommendation remains for review or proceeds automatically.

These controls reduce obvious duplication but are not a complete consent or compliance system. Keep authoritative consent and exclusion data in the organization's approved process and test how it maps to outreach eligibility.

Approve or reject an action

Review the target record, proposed channel, content, sender, timing, and reason. Approving authorizes the next processing stage; it does not prove delivery. Rejecting should preserve the recommendation outcome for later evaluation.

If edits are required, change the source or proposal through the supported workflow, then re-preview it. Avoid approving one proposal and manually sending a materially different communication outside the audit trail.

Autonomous execution

Enable autonomous execution only for reversible, well-tested actions with reliable inputs. Start with a small scope and monitor false positives, duplicate suppression, provider failures, customer replies, and actual outcomes.

RAIN OS does not guarantee a fully closed-loop campaign. Operators remain responsible for reviewing exceptions, stale records, unhandled replies, and actions that could not complete.

Finish and reconcile

After release, compare the intended audience with generated members, recommendations, dispatched work, provider outcomes, replies, and follow-up records. Assign every exception an owner and next step. Close the run only when unexplained inclusions, exclusions, duplicate work, stale recommendations, and unhandled replies have been resolved or deliberately deferred.

Troubleshooting

Expected records are missing: Review list filters, ownership and sharing, contact data, consent, recent outreach, replies, and schedule eligibility.

Unexpected records are included: Pause release, inspect the filter and suppression inputs, and correct the list before dispatch.

A recommendation does not execute: Check approval state, confidence and action settings, due time, feature enablement, required sender and recipient, and asynchronous failures.

Duplicate contact occurred: Stop the batch, compare communication records and provider references, then review claim, retry, and suppression timing before resuming.

Delivery succeeded but no follow-up appears: Check whether the provider supplied the expected callback and whether the reply or post-delivery process is enabled for that channel.

RAIN OS overview and boundaries

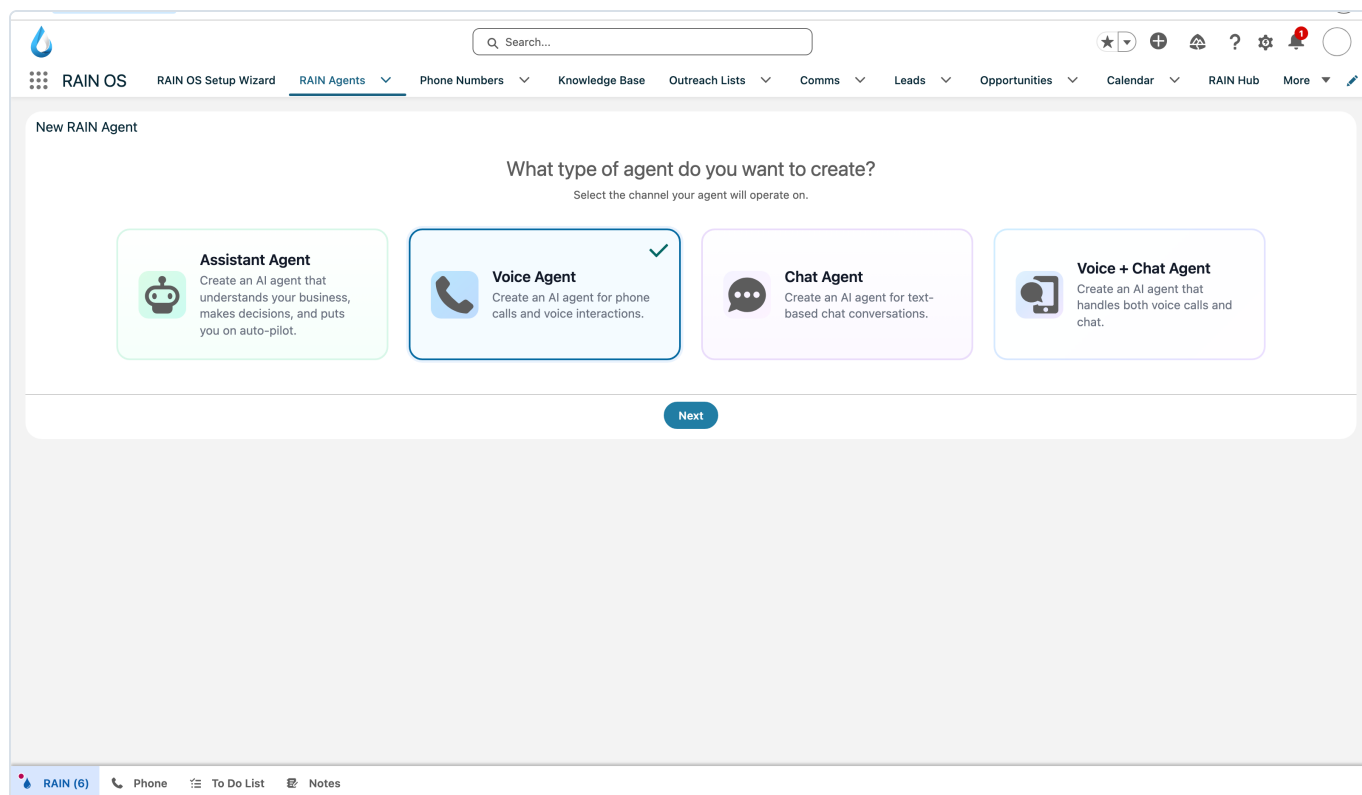
Understand what the Salesforce-native RAIN OS package provides on its own and where the optional RAIN Intelligence service begins.

What RAIN OS is

RAIN OS is a managed, Salesforce-native application for configuring agents, communication channels, outreach, suggestions, insights, scoring, and Salesforce automation. Its records, Apex services, scheduled work, reports, and primary user interfaces run in Salesforce.

The package has no declared package dependency. Install RAIN OS before any optional product that integrates with it. A customer can use RAIN OS without connecting RAIN Intelligence, subject to configuring the Salesforce-native features and any communication or AI providers those features require.

Standalone capabilities



The native Salesforce app exposes setup, agents, phone, knowledge, outreach, communications, records, and Hub work areas. The selected state shows that agent type and channel scope are defined inside RAIN OS.

RAIN OS owns the Salesforce records and workflows for these capability areas:

- Agent definitions, versions, tools, assignments, and publication lifecycle.
- SMS, voice, chat, and related communication activity.
- Phone-number and voice-agent administration.
- Outreach lists and next-best-action review.
- Insights, suggestions, and RAIN scoring.
- Invocable actions and scheduled Salesforce automation.
- Lightning application surfaces, record pages, reports, and a usage dashboard.

Some capabilities call external providers. "Salesforce-native" describes where the application and operational records live; it does not mean that every message, call, or model operation is performed entirely inside Salesforce.

The optional RAIN Intelligence boundary





Connect to Salesforce
Authorize RAIN Intelligence to access your Salesforce org

Environment

Production
login.salesforce.com

Sandbox
test.salesforce.com

RAIN Intelligence will have access to:

- ✓ Read and query your data
- ✓ Access metadata and schema
- ✓ Create, update, and deploy (when requested)

Want RAIN inside Salesforce?
Install the native RAIN OS package after connecting, or install it first if you are the Salesforce admin.

Install RAIN OS Package

Connect with Salesforce →

You can revoke access anytime from your Salesforce settings

RAIN Intelligence begins at a separate hosted Salesforce connection step. The screen also keeps native RAIN OS package installation distinct from hosted access.

RAIN Intelligence is a separately operated hosted service. It adds persistent chat, multi-model routing, retrieval, generated artifacts, gated hosted tools, background analysis, and vertical intelligence. It does not install additional Salesforce metadata; RAIN OS remains the Salesforce foundation and prerequisite.

RAIN OS supplies the Salesforce-side permission gate, configuration, Lightning entry point, and Flow action used to reach the hosted service. Without that connection, hosted chat and hosted automation are unavailable, but the rest of RAIN OS does not become unusable.

Records and results

RAIN OS writes results to the Salesforce records that own each process. For example, communication status remains with communication activity, agent publication state remains with agent versions, and reviewed recommendations remain with suggestion or insight records. Provider identifiers and error details may also be retained to support reconciliation.

Use Salesforce sharing, field access, audit history, reports, and operational status fields to understand who can see a result and whether a process completed. A successful UI submission means the request was accepted; asynchronous work or an external provider may still be processing it.

Choosing the right documentation family

Use the RAIN OS articles for Salesforce installation, permissions, agents, communications, outreach, suggestions, scoring, automation, and Salesforce UI behavior.

Use the RAIN Intelligence articles for hosted connection setup, chat, models, retrieval, artifacts, hosted voice, billing modes, the web workspace, and hosted-service health. When a workflow crosses both products, troubleshoot the Salesforce request and the hosted execution as two linked stages.

Boundary checks

Before enabling a feature, identify:

1. The Salesforce record and permission set that own the workflow.

2. Whether the workflow requires a communication provider, an AI provider, or RAIN Intelligence.
3. Which credential is used and what data that credential can access.
4. Whether the result is immediate, queued in Salesforce, or delegated to a provider.
5. Where users and administrators will see completion or failure.

This prevents a hosted-service problem from being treated as a package problem, and it prevents a Salesforce access problem from being treated as a model problem.

Phone, voice, and post-call processing

Connect phone numbers to voice agents, place or schedule calls, and verify provider, communication, task, transcript, and post-call results as separate stages.

Voice lifecycle

The screenshot shows the 'New RAIN Agent' setup interface in the RAIN OS. The top navigation bar includes 'RAIN OS', 'RAIN OS Setup Wizard', 'RAIN Agents' (selected), 'Phone Numbers', 'Knowledge Base', 'Outreach Lists', 'Comms', 'Leads', 'Opportunities', 'Calendar', 'RAIN Hub', and 'More'. A search bar is also present. The main content area is titled 'New RAIN Agent' and asks 'What type of agent do you want to create?' with the instruction 'Select the channel your agent will operate on.' Below this are four options:

- Assistant Agent**: Create an AI agent that understands your business, makes decisions, and puts you on auto-pilot.
- Voice Agent** (selected): Create an AI agent for phone calls and voice interactions.
- Chat Agent**: Create an AI agent for text-based chat conversations.
- Voice + Chat Agent**: Create an AI agent that handles both voice calls and chat.

A 'Next' button is located at the bottom of the selection area. The bottom status bar shows 'RAIN (6)', 'Phone', 'To Do List', and 'Notes'.

Choosing Voice or Voice + Chat establishes the channel scope that number mapping, safeguards, and post-call processing must support.

RAIN OS manages Salesforce phone-number records and connects them to supported voice agents. An outbound call can be started by a user action, a Flow action, or scheduled communication automation.

The lifecycle has distinct stages:

1. Salesforce validates the number, agent, recipient, and timing.
2. The voice provider accepts or rejects the call request.

3. Provider callbacks update ringing, connected, ended, or failed state.
4. RAIN OS records available duration, cost, outcome, sentiment, recording, transcript, and summary details.
5. Related Salesforce activity is created or updated.
6. Post-call analysis may create insights, suggestions, or coaching results asynchronously.

Not every provider or call supplies every result. Missing sentiment, recording, or transcript is not by itself proof that the call failed.

Prerequisites

- Configure the supported voice provider and verify account status.
- Synchronize, purchase, or connect a phone number through the supported administration flow.
- Attach the number to a published voice-agent version.
- Confirm calling permissions, recipient consent, recording notice, and regional requirements.
- Ensure the runtime user can create communication and activity records.
- Configure post-call processing only after basic call status works reliably.

Configure a phone number

1. Synchronize available numbers from the provider or complete the supported purchase flow.
2. Review number capabilities and ownership.
3. Select the published voice agent that should answer or place calls.
4. Configure supported inbound, outbound, transfer, and add-on behavior.
5. Save and verify remote synchronization.
6. Place a low-risk test call and inspect the resulting Salesforce records.

Changing the linked agent or provider options may require synchronization before the new behavior is live.

Place or schedule an outbound call

Provide the target record, valid recipient number, voice agent or phone number, and any supported call variables. For scheduled calls, also provide a future time in the intended Salesforce time-zone context.

After submission, monitor the communication record. A queued result means the call has not completed. Do not infer the business outcome until the provider sends the terminal status and post-call processing finishes.

Retry or redial behavior must be configured conservatively. Confirm the first attempt is terminal before manually starting another call, and avoid retrying validation, consent, or invalid-number failures.

Post-call results

RAIN OS can retain provider-supplied call metadata and use the transcript or summary for follow-up analysis. Related activity gives Salesforce users a familiar timeline entry, while the communication record remains the channel-specific source for status and provider details.

Recordings and transcripts can contain sensitive personal or regulated data. Restrict their visibility, follow the organization's retention policy, and confirm the provider's storage terms.

Post-call insights and suggestions are recommendations. Review them according to the organization's approval and autonomy settings before relying on them for customer-visible changes.

Troubleshooting

A number does not appear: Recheck provider authorization, number ownership, synchronization status, and user access.

The call is rejected immediately: Validate number format, published agent, provider balance, allowed destination, consent, and required variables.

The call ended but no Salesforce activity exists: Check callback processing, communication-record errors, asynchronous jobs, and activity creation permissions.

There is no transcript or recording: Confirm the provider supplied it, recording was enabled and lawful, the call connected long enough, and callback processing finished.

Post-call analysis is missing: Verify the call reached an eligible terminal state, transcript or summary input exists, analysis is enabled, and the asynchronous job did not fail.

RAIN Intelligence compatibility

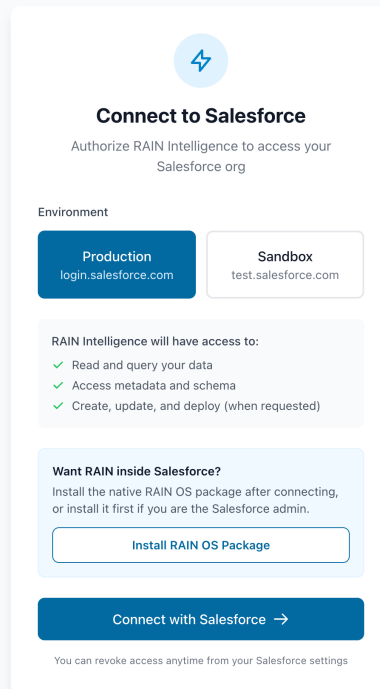
Connect the optional hosted add-on only after RAIN OS access, service configuration, user entitlement, and a low-risk end-to-end request have been verified.

Optional add-on boundary

RAIN Intelligence is not required for standalone agents, communications, phone, outreach, suggestions, scoring, automation, or reporting provided by RAIN OS. It is required for hosted persistent chat, multi-model routing, retrieval, generated artifacts, hosted gated actions, and hosted vertical intelligence.

RAIN OS owns the Salesforce entry point and connection settings. The hosted service owns model execution, hosted storage, retrieval, and web workspace behavior.

Prerequisites



The image shows a 'Connect to Salesforce' dialog box. At the top is a blue circular icon with a white lightning bolt. Below it, the title 'Connect to Salesforce' is followed by the text 'Authorize RAIN Intelligence to access your Salesforce org'. Under the heading 'Environment', there are two buttons: 'Production' (with 'login.salesforce.com' below it) and 'Sandbox' (with 'test.salesforce.com' below it). A section titled 'RAIN Intelligence will have access to:' lists three permissions with green checkmarks: 'Read and query your data', 'Access metadata and schema', and 'Create, update, and deploy (when requested)'. Below this is a light blue box titled 'Want RAIN inside Salesforce?' containing the text 'Install the native RAIN OS package after connecting, or install it first if you are the Salesforce admin.' and an 'Install RAIN OS Package' button. At the bottom is a large blue 'Connect with Salesforce →' button. A small footer note states 'You can revoke access anytime from your Salesforce settings'.

The hosted connection surface distinguishes Production from Sandbox and states its broad access categories. Compatibility still depends on entitlement, supported package versions, permissions, and backend health.

- Install and initialize RAIN OS first.
- Obtain the appropriate RAIN Intelligence subscription or entitlement.
- Establish the supported hosted connection for the Salesforce org.
- Configure the Salesforce-side service connection through the supported administration surface.
- Assign the RAIN Intelligence custom-permission set only to intended users.
- Ensure the connected Salesforce identity has least-privilege access to required records and metadata.
- Review hosted data handling, model providers, billing mode, and safety gates.

Compatibility validation

1. Confirm ordinary RAIN OS pages and records work without relying on the hosted service.
2. Open the RAIN Intelligence entry point as an entitled standard user.

3. Confirm the hosted service identifies the intended Salesforce org and user context.
4. Run a read-only request against a non-sensitive test record.
5. Verify the answer is grounded in the expected Salesforce data.
6. Run a gated low-risk action and confirm the approval behavior.
7. If Flow integration is required, test synchronous and queued behavior separately.
8. Review hosted activity, cost, and failure telemetry for the test.

Do not begin with a deployment, bulk update, customer communication, or broad record analysis.

Access model

The RAIN Intelligence permission set gates the Salesforce UI. It does not grant hosted licensing, create a connection, or broaden Salesforce record access.

Hosted Salesforce operations run through the connected credential available to the service and through RAIN safety policy. Administrators must scope that connected credential appropriately; do not assume every operation automatically has exactly the interactive user's Salesforce permissions.

Flow compatibility

The Run RAIN Intelligence action supports synchronous and background requests. Synchronous mode can return a hosted result to the Flow. Background mode queues Salesforce work and does not return the completed result to the original Flow or automatically update its source record.

Design Flow behavior around the actual response mode. A queue reference is useful for tracing but is not a completed analysis.

Upgrade and incident behavior

RAIN OS and RAIN Intelligence can release independently. After either product changes authentication, request schema, gate behavior, models, or result structure, repeat the compatibility validation.

If RAIN Intelligence is unavailable, pause hosted-dependent automation or route it to review. Do not disable unrelated standalone RAIN OS workflows unless they share a failing provider or configuration.

Troubleshooting

The tab is unavailable: Check the RAIN Intelligence permission set, app navigation, and user entitlement.

The tab opens but does not connect: Check org registration, service configuration, hosted health, and connection authorization.

A read returns no records: Verify the intended org, connected credential, object and field access, sharing, and query scope.

An action is blocked: Review safety-gate mode, action risk, required confirmation, Salesforce validation, and credential permissions.

A Flow queues but no result appears: Queued mode does not update the record with a completed result. Use supported synchronous persistence or redesign the follow-up stage.

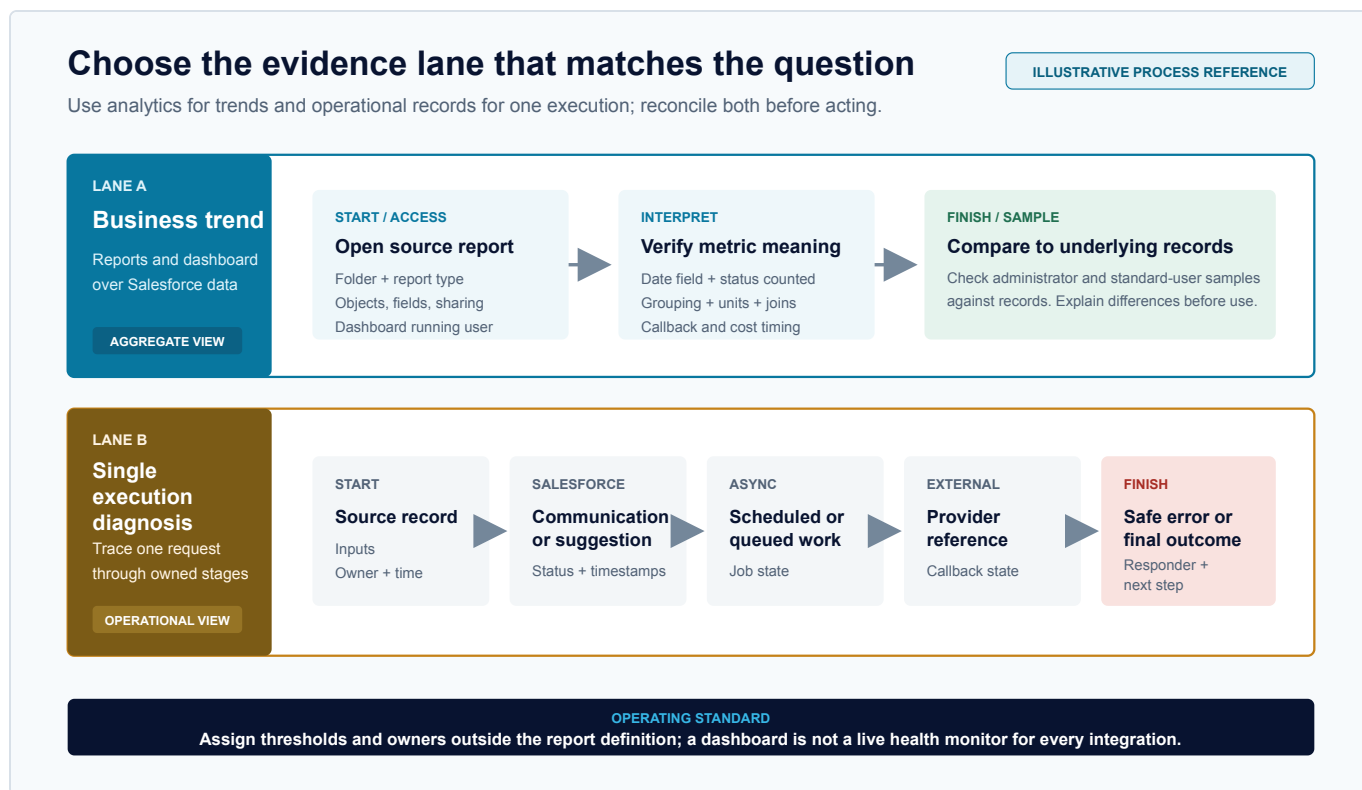
Reports, dashboard, and observability

Use packaged analytics for business trends and Salesforce operational records for execution diagnosis, while accounting for report filters, access, and provider timing.

Analytics and observability are different

RAIN OS includes packaged reports across agents, communications, delivery, cost, conversion, outreach, insights, suggestions, and next-best actions, plus a communication-usage dashboard. These answer business and operating questions over Salesforce data.

Troubleshooting a single execution requires operational evidence instead: the source record, communication or suggestion status, timestamps, asynchronous job, provider reference, and safe error detail. A dashboard is not a live health monitor for every integration.



Use reports and dashboards to interpret trends, but diagnose a single execution from its operational evidence and reconcile conclusions with source records.

Report access

Users need access to the report or dashboard folder, underlying report type, objects, fields, and records. Salesforce's report-building guidance [🔗](#) and dashboard creation guidance [🔗](#) provide the platform steps for adapting packaged analytics. Dashboard results can also depend on its running-user configuration [🔗](#). Two users can see different detail records even when a shared aggregate is visible.

After installation or an access-model change:

1. Assign folder access to the intended groups.
2. Confirm the running-user choice and sharing implications.
3. Open representative source reports as an administrator and a standard user.
4. Verify report filters, date fields, statuses, and units.
5. Compare a small sample with the underlying Salesforce records.

Interpret communication metrics

Communication totals can differ depending on whether the report counts created, queued, sent, delivered, connected, or completed records. Provider callbacks may arrive after the reporting period or update a record later. Scheduled work can be created in one period and delivered in another.

Cost fields depend on provider-supplied or calculated data and may be absent for some channels or failed requests. Use them for operating visibility and reconciliation, not as the sole billing ledger.

Interpret recommendation metrics

Suggestion and next-best-action reports can distinguish generated, viewed, approved, rejected, applied, and failed outcomes. A high approval rate is not enough to establish quality; review whether approved actions completed and produced the intended business result.

Insight counts can reflect generation volume rather than unique business events. Check grouping and duplicate rules before comparing teams or time periods.

Monitor operations

Establish a routine that covers:

- Failed or long-pending communications.
- Provider callback and reconciliation gaps.
- Failed asynchronous and scheduled jobs.
- Suggestions that were approved but not applied.
- Stale scores or delayed refresh work.
- Agent versions that are locally active but not synchronized.
- Credit, limit, or authorization warnings.

Record thresholds and ownership outside the report definition so an alert has a clear responder and escalation path.

Finish the review

For a trend review, save the effective filters and date basis, compare a small record sample, and record the business interpretation. For an incident, trace one affected execution through its Salesforce, asynchronous, and provider stages until a final outcome or owned next step is documented. Do not close either review on a dashboard total alone.

Troubleshooting analytics

A report is empty: Check date and status filters, folder and object access, record sharing, and whether the expected process writes that object.

Totals differ from a list view: Compare filters, date fields, report type joins, grouping, and record visibility.

The dashboard is stale: Refresh it and inspect source reports. A refreshed dashboard can still lag an external provider whose callback has not arrived.

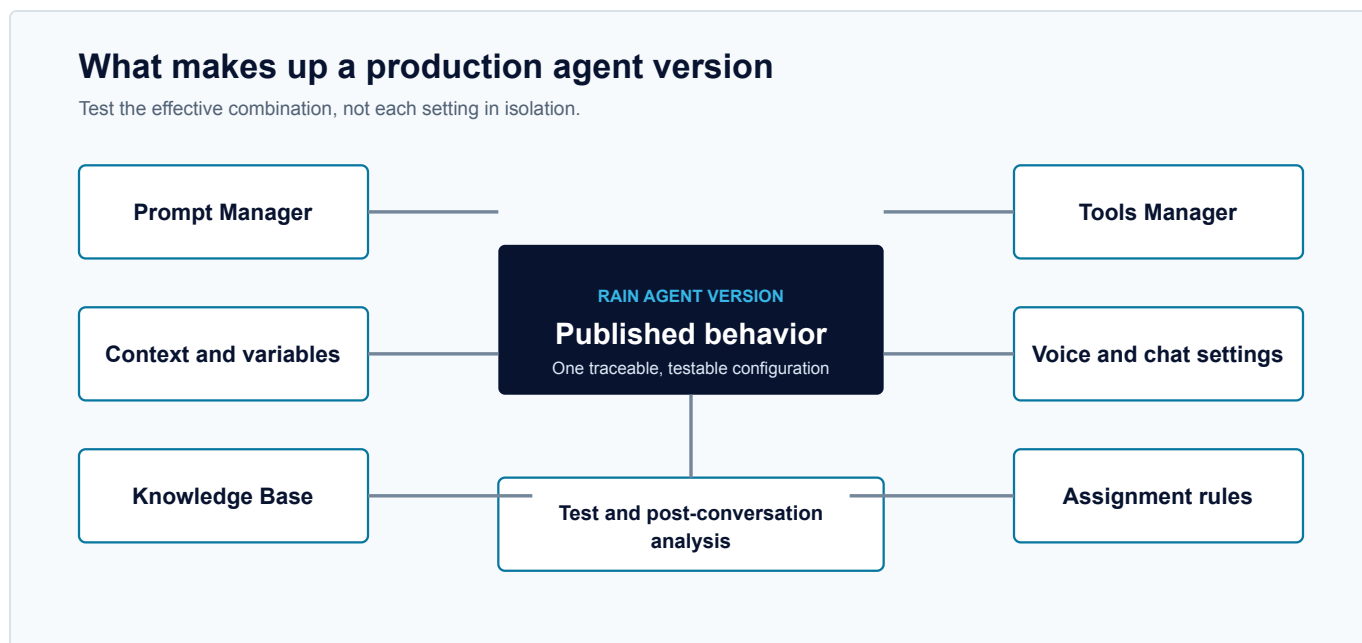
Cost is missing: Check whether the provider supplies cost for that event and whether the call reached a billable stage.

A report changed after upgrade: Review field and status changes, subscriber customizations, and the effective report definition before restoring an older copy.

Agent tools, knowledge, and actions

Attach only validated tools and relevant knowledge to an agent version, then test permissions, provider synchronization, inputs, outputs, and failure handling together.

Tool and knowledge roles



Knowledge grounds responses, while tools perform bounded operations; both are versioned dependencies that must be validated together.

Tools let an agent perform a bounded operation, such as reading a Salesforce record, invoking a Flow or Apex action, sending a communication, checking availability, booking an appointment, transferring or ending a call, or extracting a structured value.

Knowledge gives the agent approved reference material. RAIN OS can associate Salesforce-managed content with an agent version and synchronize supported material to an external agent provider. This provider-backed knowledge path is distinct from the hosted retrieval system in RAIN Intelligence.

Select tools deliberately

The current catalog is driven by packaged tool templates. It includes record access, Flow and Apex or Salesforce actions, email and SMS, calendar availability and appointment booking, value extraction, and call control or transfer. The list shown in Tool Builder depends on release metadata, agent type, provider support, version state, and configuration. Treat the current menu as authoritative.

For each tool, document:

- The business purpose and allowed users.
- Required inputs and where they come from.
- The output or Salesforce record change.
- The execution identity and required permissions.
- Whether the action is read-only, reversible, customer-visible, or financially sensitive.
- Expected provider, Flow, validation, and limit failures.

Avoid adding broad tools "just in case." A smaller tool set improves predictable selection and reduces the number of actions that require elevated access.

Configure an action tool

1. Confirm the target Salesforce action or Flow is active and has a stable input/output contract. Salesforce's Flow Builder introduction [🔗](#) explains versions, activation, inputs, and debug behavior.
2. Create the agent tool with a user-readable purpose and precise input descriptions.
3. Map only required values. Define behavior for missing record context.
4. Attach the tool to a draft agent version.
5. Test success, missing input, access denied, validation failure, and provider timeout.
6. Confirm the result is written or returned where operators expect it.
7. Publish or synchronize the agent version.

Changing a Flow variable, action signature, required field, data type, or active version can break an existing agent tool even when the agent configuration itself did not change. Client developers exposing Apex should review Salesforce's [InvocableMethod](#) action guidance [🔗](#); the method still needs a RAIN tool definition and runtime authorization.

Configure knowledge

Use current, approved, scoped material. Remove duplicates and obsolete versions before linking content. For customer-facing agents, review whether the material contains internal-only guidance, personal data, contractual restrictions, or instructions that conflict with the agent's allowed actions.

The knowledge lifecycle is:

1. Add or select the Salesforce content.
2. Link it to the intended agent or version.
3. Start provider synchronization where required.
4. Wait for a usable synchronization state.
5. Test questions that should be answered, should be declined, and should return no result.
6. Re-synchronize after replacing or materially changing content.

Deleting a local link and deleting remote provider content are separate concerns. Confirm both when retiring sensitive material.

Permissions and data handling

The user configuring a tool needs access to its definition. The runtime also needs authorization for the tool, action or Flow, records, fields, provider, and downstream system. A visible tool is not proof that every runtime principal can execute it. Test with the identity that will initiate the production path.

Knowledge content may leave Salesforce when synchronized to an external provider. Review provider retention and access terms before enabling that path.

Do not use knowledge as a substitute for record-level authorization. An agent should retrieve a live Salesforce fact through an authorized tool when the answer depends on current restricted data.

Troubleshooting

The agent never selects a tool: Make the tool purpose and inputs unambiguous, confirm it is linked to the active version, and verify the channel supports it.

A tool is selected but fails: Check required inputs, Flow or action activation, field access, sharing, validation rules, callout configuration, and asynchronous limits.

Knowledge answers are stale: Confirm the current file is linked, the remote synchronization finished, and the active version uses the updated knowledge.

Knowledge produces no answer: Check file support, content readability, synchronization state, question terminology, and provider availability.

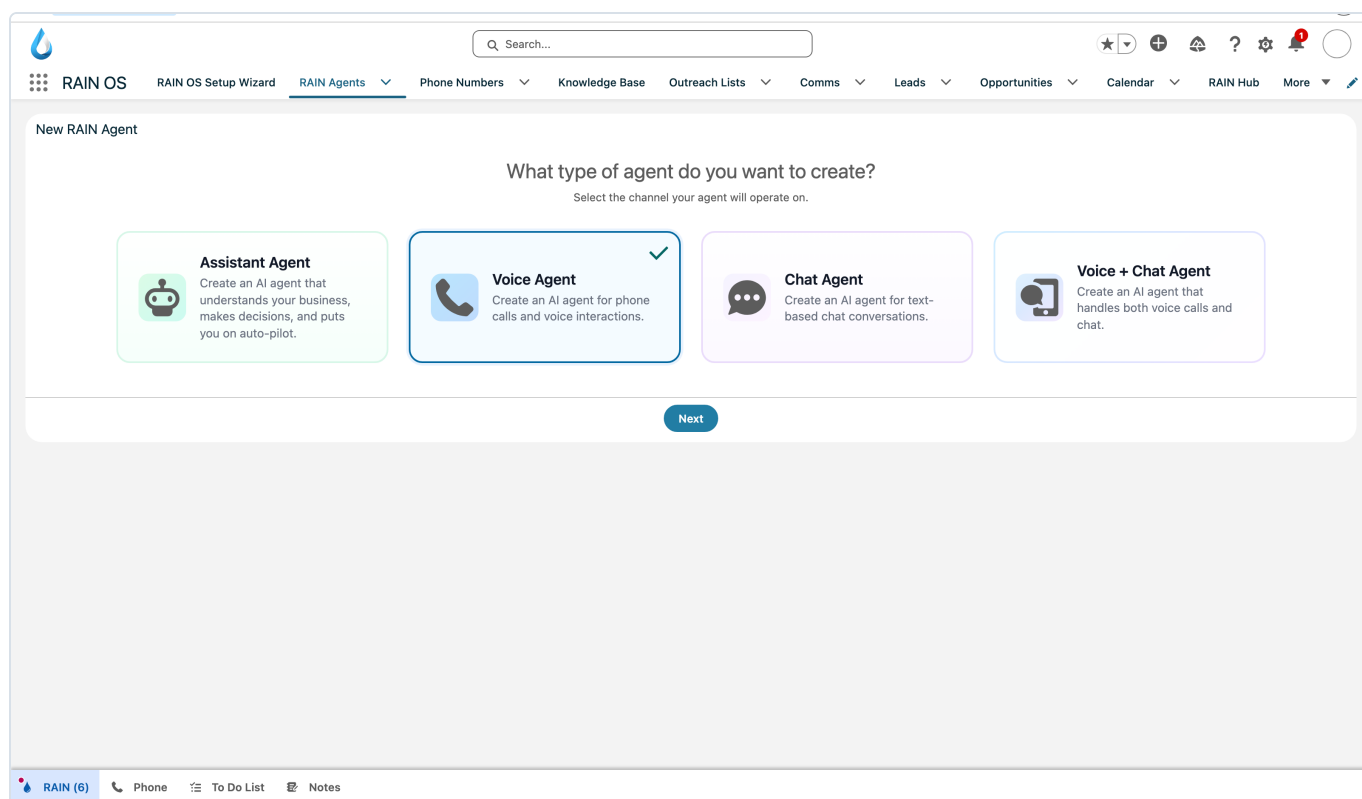
A published version still has old tools: Synchronize the version and verify the assignment or channel points to that version.

USER EXPERIENCE

UI surfaces and navigation

Find the RAIN OS application, workspaces, record pages, composers, setup tools, and utility items while accounting for permission and activation differences.

RAIN OS application



The app navigation is visible across the top, while the RAIN, Phone, To Do List, and Notes utilities remain available at the bottom. Items hidden under More depend on the assigned app configuration.

The packaged Lightning application brings together setup, agents, phone numbers, knowledge, outreach, communications, insights or suggestions, and the optional RAIN Intelligence entry point. It also includes common Salesforce records used in those workflows.

The utility bar provides persistent access to selected productivity and operational tools, including the RAIN Hub and communication-related utilities. Exact visibility depends on permission

assignment, app configuration, form factor, and subscriber Lightning activation.

Main work areas

Setup is for administrators configuring features, providers, permissions, metadata synchronization, scoring, and hosted compatibility.

Agents is for agent definitions, versions, tools, knowledge, assignments, publication, and channel conversion.

Communications and phone is for senders, phone numbers, message composition, calls, communication history, and delivery status.

Outreach and recommendations is for outreach lists, next-best actions, insights, suggestions, and review decisions.

RAIN Hub presents operational work and shortcuts. Treat the underlying Salesforce records as the authoritative status when a compact surface omits detail.

RAIN Intelligence opens the optional hosted experience for entitled and connected users.

Record pages and actions

RAIN OS packages Lightning record pages for core agent, phone, outreach, insight, suggestion, and communication work. Quick actions support frequent operations such as cloning or publishing an agent, converting a channel, managing a number, placing a call, or opening a widget.

A packaged page is not necessarily the active page for every app, profile, record type, or device. Administrators should review Salesforce's Lightning page activation guidance [🔗](#) after installation and upgrades, particularly when the subscriber already has custom record pages.

Communication composers

RAIN OS provides channel-specific and combined composer experiences. The available controls depend on record context, provider configuration, sender availability, user access, and the selected channel.

When a composer opens without expected values, check the source record and permissions before editing the component. When it submits successfully, continue to the communication record to monitor asynchronous status.

Configure visibility

1. Assign the appropriate RAIN OS permission set using Salesforce's permission-set assignment steps .
2. Add the RAIN OS app to the user's available applications.
3. Confirm required tabs are visible in the app navigation.
4. Activate packaged or subscriber record pages for the intended app, profile, record type, and device.
5. Review component visibility rules and custom-permission conditions.
6. Test as a standard user with representative record sharing.

Avoid giving broad administrative access solely to make a tab appear. Resolve navigation, object access, and component visibility at their correct layer.

Empty, loading, and error states

An empty list can mean there are no records, the user lacks sharing access, a filter excludes records, or synchronization has not completed. A loading state can represent an Apex request or provider operation. An error should be correlated with the affected record, asynchronous job, or provider status.

If a component repeatedly fails, capture the user role, app, page, record type, record state, action, and time. Do not include credentials or sensitive message content in support evidence.

Troubleshooting navigation

The app is missing: Check permission-set assignment, app availability, license, and whether the user is in the intended Salesforce org.

A tab is missing: Check tab visibility and app navigation independently from object access.

A record uses an older layout: Review Lightning page activation and record-type assignment.

An action is missing: Check page layout action placement, component visibility, record state, and custom permissions.

The RAIN Intelligence tab opens but cannot connect: Follow the hosted compatibility checklist; the tab permission alone is not a hosted connection.