



SOURCE GROUNDED AUDIT

SymposiumDemo workspace audit: risks, conflicts and what to fix first

October 18, 2026 · 15 verified sources

The static scan flagged 30 signals, but the biggest problems are logic conflicts between automations. The cancellation flow issues credits that a validation rule rejects for 88 active bookings, and the price lock covers 1 of 1,627 bookings. Org evidence shows the permission findings are low risk here. Fix first: the credit reason, fault paths on the flows that write records, and the quoted-price backfill.

Summary and fix-first list

The static scan reported 30 signals (1 critical, 9 high), all of them permission or code-hygiene patterns. Reading the source and the org evidence changes the picture. **The biggest problems are logic conflicts between automations that the scanner cannot see.** Almost all of the permission findings turn out to be low risk in this org. Evidence labels: **[1]** is workspace metadata read for this audit, **[2]** is read-only SOQL and Tooling API results from the org, and the other markers are Salesforce documentation passages.

ACTIVE BOOKINGS THAT WOULD FAIL TO CANCEL

88 of 1,626

Half-value credit would exceed \$500 with no reason; largest booking is \$1,500 [2]

STATIC SIGNALS AFTER RE-RATING

1 high

5 medium, 22 low, 2 dismissed as false positives (analyst re-rating of the 30 signals)

BOOKINGS COVERED BY THE PRICE LOCK

1 of 1,627

Quoted_Price__c is blank on the other 1,626 [2]

PROJECT COMPONENTS THAT FAILED TO RETRIEVE

0

227 distinct failures, none project-owned; the reported 454 double-counted [2]

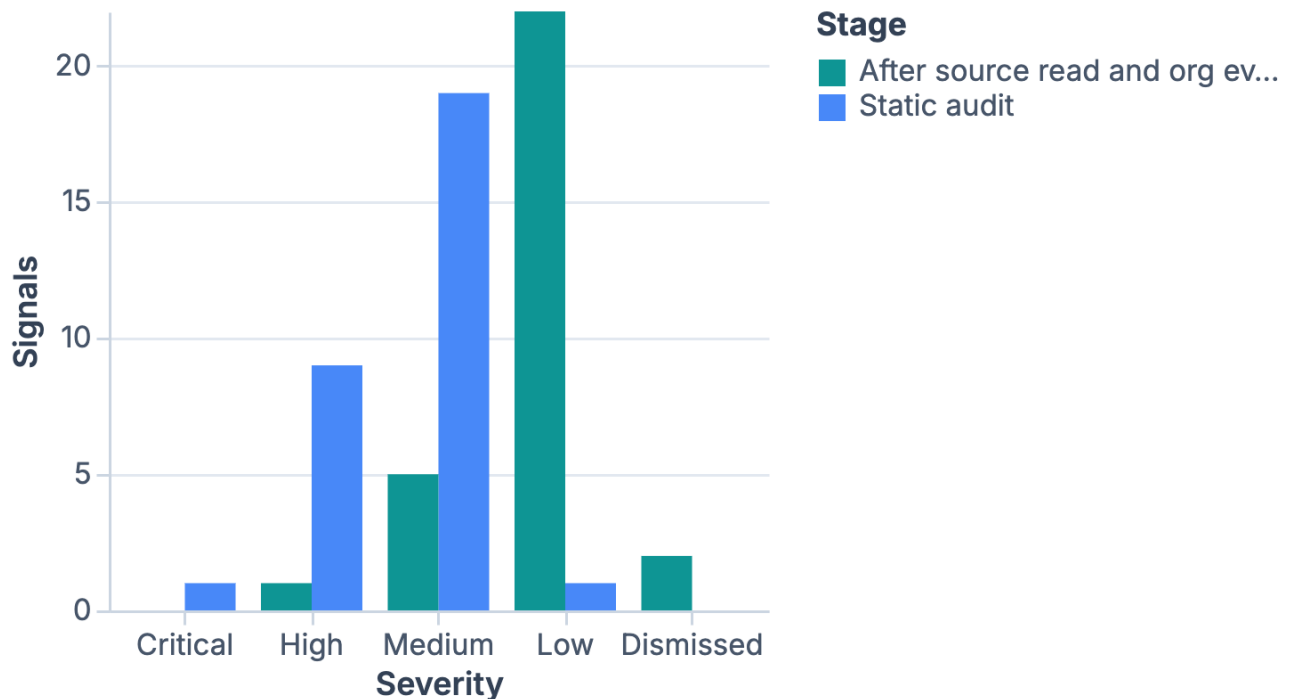
Fix first

Risk: Cancelling a booking worth more than \$1,000 should fail today. The cancellation flow issues a credit of half the booking value without a reason, and a validation rule rejects any credit over \$500 that has no reason. The failing step has no fault path, so the user is likely to see a generic flow error. Only one booking has ever been cancelled, so nobody has hit this yet. The failure has not been reproduced, because reproducing it means cancelling a booking and this audit is read-only.

#	What to fix	Why it ranks here	Evidence	Rough effort
1	Make flow-issued credits valid: pass a reason into Issue_Resort_Credit (or exempt flow-issued credits from the rule). Do the same for Issue_Bulk_Resort_Credits.	Breaks cancellation for about 5% of active bookings; the agent bulk-credit action has the same flaw above \$500	Booking_ Cancellation. flow- meta. xml:181- 202; Issue_ Resort_ Credit. flow- meta. xml:28- 48; Large_ Credit_ Needs_ Reason. validationRule- meta. xml:5- 6 [1]; 88 bookings [2]	Small
2	Add fault paths to the flows that write records: Issue_Resort_Credit, Issue_Bulk_Resort_Credits, Booking_ After_ Save_ Welcome_ Task, Booking_Change_Subscriber, and the lookup and loop update in Booking_Cancellation.	An unhandled fault in a record-triggered flow shows users a generic error [3]; all 11 project flows run the version in the repo, so these apply to what runs [2]	Booking_ Cancellation. flow- meta. xml:107- 160 [1]	Small

#	What to fix	Why it ranks here	Evidence	Rough effort
3	Backfill Booking__c.Quoted_Price__c, then use it in the cancellation credit formula instead of the current experience price.	The price lock exists but covers 1 booking, and the credit formula ignores it	Booking_ Cancellation. flow- meta. xml:40; BookingTriggerHandler. cls:33- 47 [1]; 1,626 blank [2]	Small to medium
4	Remove Passport_Number__c and Medical_Notes__c from the Contact_Home Data Cloud mapping, or govern the copy with data-space permissions, before anyone records a passport number.	Field-level security on Contact does not govern the copy. It holds no passport numbers today (0 populated), so this is cheap now and costly later	Contact_ Home_ map_ Contact_ Home_ 1792277138740. objectSourceTargetMap- meta. xml:88- 90, 464-466 [1]; 0 populated [2]	Medium
5	Stop DataTreeImporter from swallowing exceptions, and stop the Import button running generateData after a failed import.	A failed import still lets generateData delete every booking and session	DataTreeImporter.cls:90-92; sampleData.js:11-12; SampleDataGenerator. cls:26- 27 [1]	Small
6	Run all Apex tests once to record real coverage. Add a bulk test (20 or more bookings) and a re-activation test for BookingTriggerHandler; add a test for ContextService.	Stored coverage reads 0%, so coverage is unmeasured. The handler tests only insert single bookings	BookingTriggerHandlerTest. cls:30,38,51,62 [1]; coverage query [2]; bulk guidance [4]	Small
7	Housekeeping: bulkify ListExperiencesFromType, turn on field history for Is_Canceled__c, remove the unused http://www.apexdevnet.com remote site, review Create_Booking (runs without sharing and nothing in the repo calls it).	Low-probability or low-impact items; they should not delay 1 to 6	See the Risks page [1]	Small

Priority order after combining source reading with org evidence



The 30 static signals before and after review. The re-rating is the analyst's judgement from reading source and org evidence, not a scanner output [1][2].

Conflicts

Each conflict below is a place where two pieces of this org's automation or configuration disagree. Confidence reflects what was read versus what ran.

1. Cancellation credit vs. the large-credit validation rule

Confidence: high on the configuration, medium on the user-visible symptom.

Booking_Cancellation is an after-save flow that fires when Is_Canceled__c becomes true (Booking_Cancellation.flow-meta.xml:161-179). It calls the Issue_Resort_Credit subflow with amount = 50% x experience price x guests and a contact, and nothing else (:36-42, :181-202). The subflow creates a Credit__c from only those two inputs (Issue_Resort_Credit.flow-meta.xml:28-48), and that create has no fault path. Credit__c.Large_Credit_Needs_Reason rejects Amount__c > 500 when Reason__c is blank (Large_Credit_Needs_Reason.validationRule-meta.xml:6). The rule's own description says the credit flows do not set a reason (:5), so the conflict was known but not reconciled.

Org evidence: 88 of 1,626 active bookings have a total price above \$1,000, so their credit would be over \$500; the largest is \$1,500, which gives a \$750 credit [2]. One booking has been cancelled so far, with a \$28 credit and a blank reason, which passed [2]. Flow-interview logging is off, so earlier failures would have left no record [2].

What this likely does to a user: Salesforce's training material describes an unhandled fault in a record-triggered flow as a generic "unhandled fault" save error, and

describes fault paths on Create Records and other elements as the fix [3]. This was not reproduced here, and I have not confirmed that the failure rolls back the cancellation itself. Issue_Bulk_Resort_Credits has the same gap and no fault path (Issue_Bulk_Resort_Credits.flow-meta.xml:111-120), and the agent topic tells the agent to use it for multi-contact credits (genAiPlugins/Customer_Service_Assistant.genAiPlugin-meta.xml).

2. Credit formula vs. the price lock

Confidence: high. BookingTriggerHandler stamps Quoted_Price__c so that a later price change cannot alter the quote (BookingTriggerHandler.cls:32-47; Quoted_Price__c.field-meta.xml:4). The cancellation credit ignores it and uses Session__r.Experience__r.Price__c, today's price (Booking_Cancellation.flow-meta.xml:37-40). The org shows 1,626 of 1,627 bookings with no quote, because they predate the trigger. The single cancelled booking has a quote equal to the current price (\$55), so no money has moved on a stale price yet [2]. Today's price decides every credit, so a price rise before a cancellation increases it and a price cut decreases it.

3. Sensitive-field policy vs. the Data Cloud copy

Confidence: high on the mapping, medium on impact. Passport_Number__c and Medical_Notes__c are both described as sensitive (Passport_Number__c.field-meta.xml:4; Medical_Notes__c.field-meta.xml:4). The permission sets are designed around that: Concierge "never sees passport numbers" and Marketing has "no passport or medical access" (Concierge.permissionset-meta.xml:3; Marketing.permissionset-meta.xml:3). The Contact_Home data stream maps both fields into Contact_Home__dll (objectSourceTargetMaps/Contact_Home_map_Contact_Home_1792277138740.objectSourceTargetMap-meta.xml:464-466). CRM field-level security does not govern that copy, so the permission design is bypassed by the Data Cloud mapping.

Org evidence: no Contact has a passport number today [2]. Medical notes could not be counted (SOQL cannot filter a long-text field), and the stream's status and last refresh could not be read [2], so whether the copy is live is unverified. The Data Cloud extract permission set (sfdc_a360_sfcrm_data_extract) grants object-level View All Records and View All Fields on Contact in the metadata, and is held by the Data Cloud integration user [2]. It does not set the ViewAllData user permission, which is narrower than the packet's wording. Front_Desk, which records passport numbers at check-in, is assigned to no one yet [2]. This risk becomes real the day passport numbers start being collected.

4. Sample-data import vs. its own error handling

Confidence: high. The Import tab runs importData() and then generateData() (sampleData.js:11-12). importData catches every exception and only calls System.debug (DataTreeImporter.cls:90-92), so a failed import resolves normally. generateData then deletes every Booking__c and Session__c (SampleDataGenerator.cls:26-27), and throws if no contacts

exist. The user can see a success toast after a failed import, or an error toast after the data has already been deleted. Exposure is small: the org is a Developer Edition, not a sandbox, and only Coral_Cloud_Admin shows the Import tab, held by one administrator [2][1]. Keep it admin-only and remove it before copying the pattern into a real org.

Conflict	Source	Effect today	Rating
Cancellation_Needs_Reason validation vs. the before-save defaults flow: the flow fills a blank reason with Other, so the rule can't fire from the UI	Booking_ Before_ Save_ Defaults. flow- meta. xml:7- 17, 88; Cancellation_ Needs_ Reason. validationRule- meta. xml:5 [1]; before-save flows run immediately before Apex before triggers [5]	None: the rule is effectively dead code, and its own description says so	Low
Capacity check vs. re-activating a cancelled booking: the update path only checks capacity when guests increase	BookingTriggerHandler.cls:17- 27 [1]	No session is overbooked (minimum 6 slots free). Booking__c has no field history, so re-activations cannot be counted [2]	Low
Report-export transaction security policy is active but its flow is Draft	transactionSecurityPolicies/ sfdc_ default_ ReportExport_ Protection. transactionSecurityPolicy- meta. xml; flows/ sfdc_ default_ ReportExport_ Protection_ Flow. flow- meta. xml:73 [1]; Draft confirmed in org [2]	Not verified. Salesforce documentation on what an active policy does with a draft flow was not read. It is a platform-shipped default	Low, unverified
Booking_Change_Subscriber only notifies user owners; sessions owned by a queue produce no task and no error	Booking_ Change_ Subscriber. flow- meta. xml:5- 26 [1]	One event so far; subscriber running with 0 retries [2]	Low

Smaller conflicts and why they are not urgent

Risks

The scanner's 30 signals are re-rated below using what the source and the org showed. Namespaced packages (about 28,700 ssot Data Cloud components and others) are inventory only and were not risk-scanned.

Signal group	Count	Scanner severity	Final rating	Why
Powerful profile permissions (Admin: ModifyAllData, ViewAllData, AuthorApex, CustomizeApplication, ManageUsers; Analytics Cloud Integration User: ViewAllData)	6	1 critical, 5 high	Low	Admin profile permissions are expected. The org has 2 System Administrators, 1 Analytics Cloud Integration User and 1 Analytics Cloud Security User as active standard users [2]
Broad object access in permission sets (Coral_Cloud_Admin, Demo_Extensions_Admin, Finance)	3	High	Low	Coral_Cloud_Admin and Demo_Extensions_Admin are each held by one System Administrator; Finance is held by no one [2]

Signal group	Count	Scanner severity	Final rating	Why
SOQL in a loop: ListExperiencesFromType	1	High	Medium	Confirmed at ListExperiencesFromType. cls:10- 22. The query count per call grows with the input list; one transaction allows up to 100 SOQL queries [6]. It also sets WITH SYSTEM_MODE explicitly (:21), which skips field and object permissions [7]
Active flow without fault path	9	Medium	1 high, 3 medium, 5 low	Rating depends on whether the flow writes records. See the flow table below
Autolaunched flow mutates records	6	Medium	Low	No recursion found: Booking_Cancellation updates only bookings with Is_Canceled__c = false, which fall outside its entry condition (Booking_Cancellation. flow- meta. xml:138- 144, 167-175)
No companion test class	3	Medium	1 low, 2 dismissed	SampleBookingBatch and SampleDataGenerator are exercised by SampleDataTest.cls. ContextService has no test (a one-line method)
No explicit sharing mode: WeatherService	1	Medium	Low	No queries or DML. Salesforce documents that from API 67.0 a class without a sharing declaration runs in the current user context, and earlier versions keep the current sharing rule [8]; this class is on API 65
Debug logging: DataTreeImporter	1	Low	Medium	The debug call is not the issue; the swallowed exception beside it is (conflict 4)

Static signals, grouped, with final rating

Flow	Type	What it writes	Fault path	Rating
Issue_Resort_Credit	Autolaunched subflow	Credit__c	None	High: it is the failure point of conflict 1
Issue_Bulk_Resort_Credits	Autolaunched (agent action)	Credit__c list	None	Medium
Booking_After_Save_Welcome_Task	After-save, create	Task	None	Medium: a failure surfaces on booking creation
Booking_Change_Subscriber	Platform event	Task	None	Medium
Booking_Cancellation	After-save, update	Event publish (has fault path); same-day lookup and loop update (none)	Partial	Medium. The scanner did not flag this because the flow has one fault connector
Create_Booking	Autolaunched	Booking__c	Yes	Low (but runs without sharing, see below)

Flow	Type	What it writes	Fault path	Rating
Booking_Before_Save_Defaults, Get_Customer_From_Email_or_Membership_Number, Get_Experience_Details, Get_Sessions, PersonalizedSchedule	Before-save or read-only	No DML	None needed	Low

Flow fault-path review [1]

Loop inside the cancellation flow

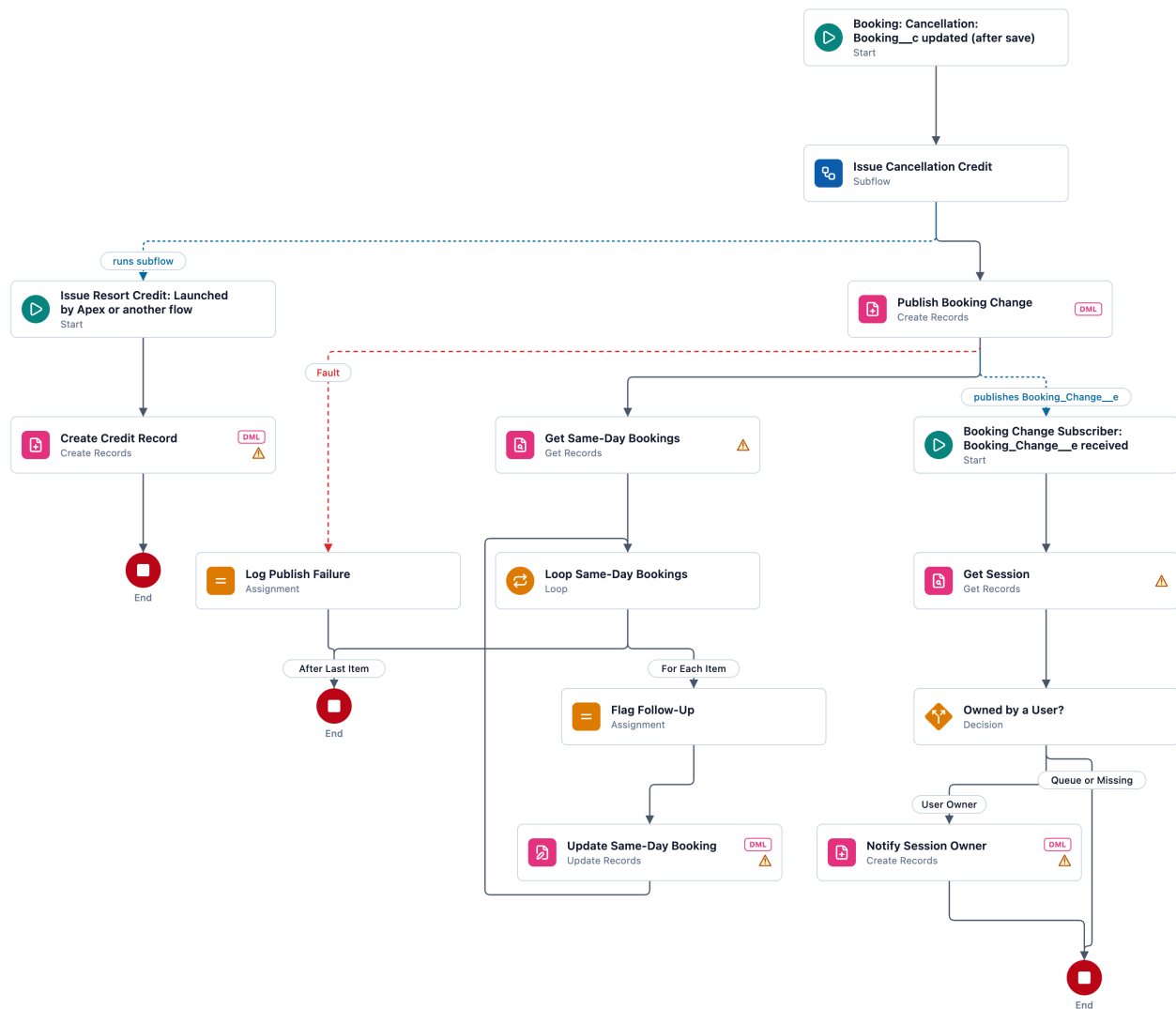
In Booking_Cancellation, the update of other same-day bookings sits inside a loop (Booking_Cancellation.flow-meta.xml:45-55, 151-160). One transaction allows up to 150 DML statements and 100 SOQL queries [6], so a guest with a very large number of same-day bookings would hit the limit. Real-world volume is probably small.

Item	Evidence	Rating
Create_Booking runs SystemModeWithoutSharing and nothing in the repo calls it (no Apex, LWC or agent function references it); all 44 profiles list it as disabled	Create_Booking.flow-meta.xml:113 [1]. Whether anything outside the repo runs it is not verifiable from metadata	Low
ApexDevNet remote site setting is active over plain HTTP and no Apex refers to it	remoteSiteSettings/ ApexDevNet. remoteSite-meta. xml:4- 5 [1]	Low
WeatherService calls an anonymous named credential (NoAuthentication) and indexes temperatures[0] and weather[0] without checking they exist	namedCredentials/ Weather_ Endpoint. namedCredential- meta. xml; WeatherService.cls:49-50, 67 [1]	Low
Source API is 65.0 while the org is on 68.0. Salesforce documents a sharing-context change at API 67.0 [8]; check BookingTriggerHandler and WeatherService when raising the project API version	sfdx-project.json; WeatherService.cls-meta.xml:3 [1]	Watch item
The role permission sets (Finance, Front_Desk, Concierge, Marketing) and the Guest_Services group are assigned to no one, so the access design is untested in practice	Org assignment query [2]	Informational

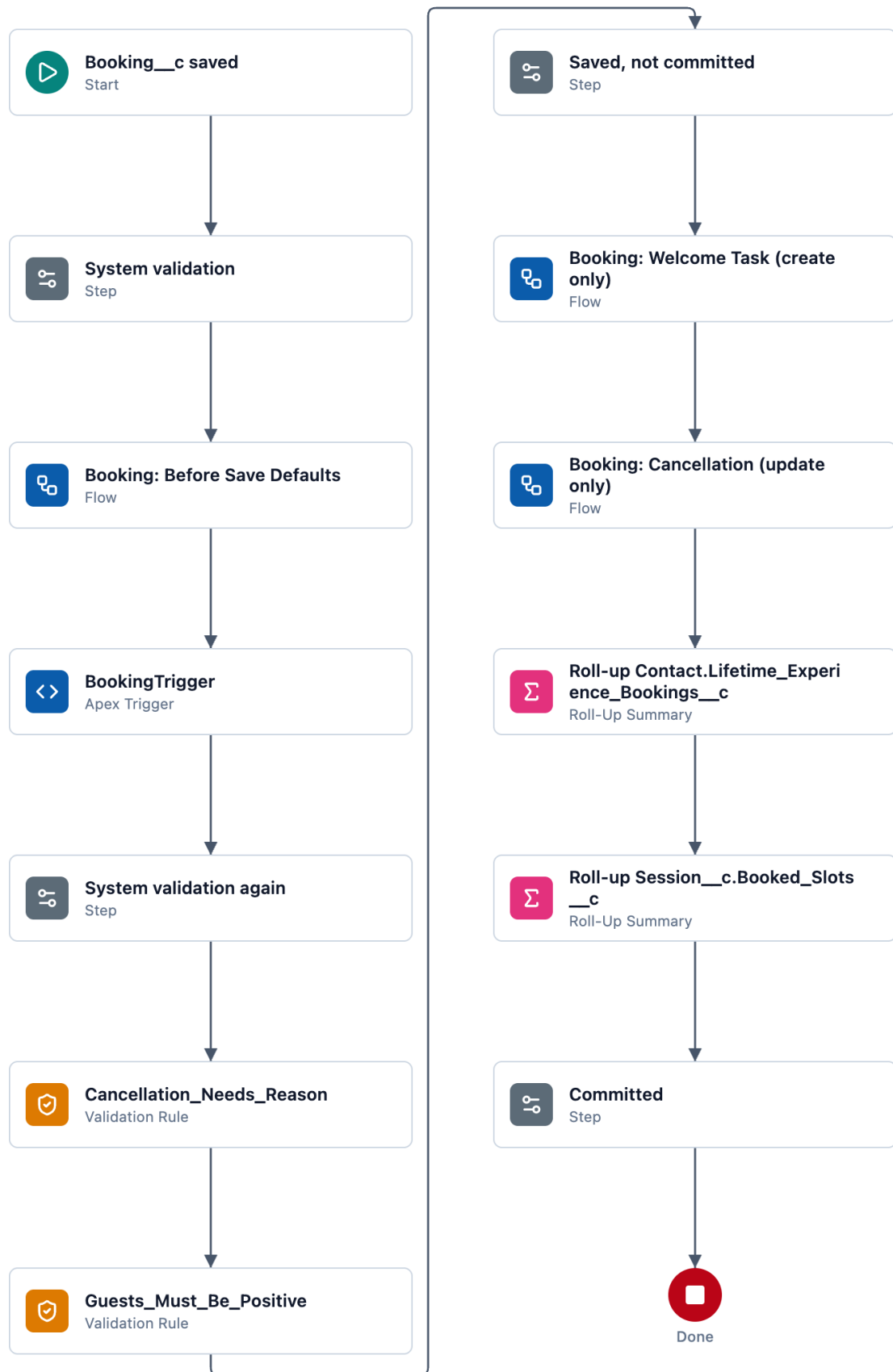
Smaller risks

Cancellation chain and save order

Both diagrams are drawn from the workspace metadata by the audit tooling. The first shows the cancellation chain with the elements that have no fault path. The second shows what runs, in order, when a booking is saved.



Booking_Cancellation, the Issue_Resort_Credit subflow it calls, and the Booking_Change_Subscriber flow started by the event it publishes. Five elements have no fault path.



Save order for Booking__c. The before-save defaults flow runs just before the Apex before trigger [5]; the welcome-task flow (trigger order 10) and the cancellation flow (trigger order 20) are after-save and run for different operations [9].

Evidence and confidence

This page keeps the kinds of evidence apart so you can see how much weight each claim carries.

Claim	Evidence type	Confidence	Notes
Cancellation credit has no reason and a rule rejects large credits without one	Workspace metadata [1]	High	Source read directly
88 of 1,626 active bookings would produce a credit over \$500	Org SOQL [2]	High (0.9)	Uses current price, the same input the credit formula uses. Not reproduced
Cancellation shows a generic flow error for those bookings	Inference from metadata plus documentation [3]	Medium	Not reproduced; rollback of the cancellation itself not confirmed
1,626 of 1,627 bookings have no quoted price	Org SOQL [2]	High (0.95)	
All 11 project flows run version 1, matching the repository	Org SOQL and Tooling API [2]	High (0.95)	So the flow findings apply to what runs
Admin permission sets are held by one System Administrator; role permission sets are held by no one	Org SOQL [2]	High (0.95)	User names were left out of the packet
No session is overbooked (minimum 6 free slots)	Org SOQL [2]	High (0.95)	Re-activations cannot be detected: history tracking is off
Booking Change subscriber is running, 0 retries, no error	Org SOQL [2]	Medium (0.85)	One event so far
Data Cloud mapping includes passport and medical fields	Workspace metadata [1]	High	Stream status and last refresh are unverified
The pull missed 227 components, none project-owned	Pull logs read by the evidence agent [2]	Medium to high (0.9)	Not read from the org; the reported 454 was double-counted
Apex coverage is 0%	Org Tooling API [2]	Low (0.7 to 0.75)	Interpreted as tests never recorded, not tests failing: 5 test results pass and none fail. Running the tests would settle it

Evidence ledger

Gaps and limits

- **Org labelling.** The packet labelled the queried org production. Per its own query the org is a Developer Edition and not a sandbox, so treat it as the live org, not as a business production system.
- **Not verified:** whether the Contact_Home stream is active and what it holds; whether any past cancellation failed (interview logging is off); whether any booking was re-activated; what an active transaction security policy does while its flow is Draft (Salesforce documentation not read); real Apex coverage.
-

Not covered: the roughly 28,700 ssot Data Cloud components and other installed-package metadata are inventory only, with no risk rules applied.

- **Documentation caveat.** The Salesforce page on Apex access modes is flagged as describing preview features, so its statement about system mode is cited with lower confidence [7].
- **Unread documentation.** The Salesforce Help article on customizing what happens when a flow fails was found but not read, so it is not cited.
- **Method.** The 30 static signals were re-rated by reading the source, so the final ratings are analyst judgement.

Sources and Citations

[1] SymposiumDemo workspace metadata, retrieved 2026-10-17 (source API 65.0, org API 68.0) Uploaded demo-salesforce-metadata.zip. File references in this report are relative to force-app/main/default/.

[2] Read-only org evidence packet 20261018T014705Z-audit-org-evidence (SymposiumDemo, Developer Edition), imported by the user. Internal findings packet imported from demo-salesforce-insights.zip; SOQL and Tooling API results run against SymposiumDemo on 2026-10-18.

[3] Handle Flow Errors Effectively | Fault Path Tips. Trailhead, Flow Implementation II.

<https://trailhead.salesforce.com/content/learn/modules/flow-implementation-2/handle-flow-errors-with-fault-paths>

[4] Testing Best Practices. Apex Developer Guide, Winter '27 (API 68.0).

https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_testing_best_practices.htm

[5] Before-Save Record-Triggered Flows. Salesforce Help.

https://help.salesforce.com/s/articleView?id=platform.flow_concepts_trigger_record.htm&language=en_US&type=5

[6] Running Apex within Governor Execution Limits. Apex Developer Guide, Winter '27 (API 68.0).

https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_limits_tips.htm

[7] Set an Access Mode for Database Operations. Apex Developer Guide, Winter '27 (API 68.0).

https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_classes_enforce_usermode.htm

[8] Enforce Sharing Rules. Apex Developer Guide, Winter '27 (API 68.0).

https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_security_sharing_rules.htm

[9] Define the Run Order of Record-Triggered Flows for an Object. Salesforce Help.

https://help.salesforce.com/s/articleView?id=platform.flow_task_trigger_run_order.htm&language=en_US&type=5