

The Parnas Suite



CHRISTOPHER JAMES AUSTIN

The Parnas Suite

Christopher James Austin

This book is available at <https://leanpub.com/theparnassuite>

This version was published on 2026-09-20



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2026 Christopher James Austin

Contents

1. The New Field	1
2. The Backbrief	5
3. The State That Never Existed	8
4. The Horror Exhibit	12
5. How Large Is Meridian?	15
6. The Lost Context	16
7. The Paper	17
8. The Cost	18
9. What Is a Module?	19
10. Two Changes	20
11. Margaret’s Theory	21
12. The Third Design	22
13. The Module Promise	23
14. Where Meridian Was Right	24
15. The Screen	25
16. The Read Side	26
17. The Second Dashboard	27
18. The Second System	28

19. The Names	29
20. The Reviews	30
21. What Happened Next	31
Author's Note	32
Buy the book!	33

1. The New Field

The change request arrived on Tuesday morning and, according to the programme office, affected only one field. Daniel Reeves read it while waiting for the lift, then read the attachment to find out what the field was supposed to mean.

REGULATORY CHANGE: ADDRESS CLASSIFICATION

A review had found that the bank could not reliably distinguish a customer's declared residential address from a correspondence destination across its channels and downstream systems. Nor could it consistently demonstrate the evidence and effective dates supporting either. The finding had to be remediated within nine months.

The programme manager had added a reassuring note: initial estimates should focus on local implementation, since this was expected to be a low-complexity change.

By the time Daniel reached the sixth floor, twenty-three teams had been invited to the impact assessment. Fourteen were likely to be involved. The others had been invited because nobody could establish whether they could safely be left out.

The lift opened opposite the Parnas Suite. Beneath the room's name, a plaque declared **HIGH COHESION. LOW COUPLING**. Three elderly men were standing in front of it.

The broad man with heavy glasses was studying a printed copy of the security instructions. A thinner man stood with his hands behind his back, looking at the plaque. The third had copied its wording into a small notebook.

"That is closer to Constantine's vocabulary," the broad man said. "Perhaps whoever ordered the sign thought they were saying much the same thing."

"Perhaps," said the man with the notebook. "I would like to know what distinction they thought they were making."

Daniel stopped beside them and asked whether they needed help. Their temporary labels read **FRED**, **PETER** and **ED**, each followed by **EXTERNAL**

REVIEWER. He had been told the previous evening that the Meridian Steering Committee had commissioned an independent review, but no biographies had accompanied the invitation. He had expected former regulators or management consultants.

“You’re here for Meridian?” Daniel asked. “Adrian Vale is presenting the architecture at nine. I’m Daniel Reeves; I’ll be taking you through some of the implementation afterwards.”

Fred introduced himself and Peter, then turned towards the man with the notebook. “Call me Ed,” the third visitor said, closing it.

Peter asked what the system did. Daniel began to describe customer services, then stopped because that could mean either the bank’s business or its software.

“It holds customer information and supports servicing across the retail bank,” he said. “Address changes, personal details, evidence, that sort of thing. It also supplies Tax, Financial Crime and Communications.”

“Thank you,” Peter said. “That gives us somewhere to start.”

The room filled with architects and delivery leads. Representatives from Customer, Tax, Financial Crime, Communications, Analytics and the channel teams joined remotely.

At nine, Adrian Vale entered. He was Meridian’s chief architect, silver-haired and composed, with fifteen years of experience trying to extract the bank from its old customer platform. Daniel had worked for him for seven of those years. Vale had promoted him twice, and had usually been the person who could turn an argument between teams into a decision somebody could implement.

Vale shook the visitors’ hands without asking for their surnames. “It is an honour,” he said. Helen Ward, the programme director, introduced them simply as the independent reviewers, and the presentation began.

The first slide showed a red cylinder labelled **CUSTOMER ORACLE DATABASE**. Branches, online banking, call centres, reporting and more than twenty internal systems surrounded it. Every arrow pointed towards the database.

“This is Atlas,” Vale said. “Meridian was created to replace it. The difficulty wasn’t merely its age. We had uncontrolled dependencies, broad access and releases in which nobody could establish the complete impact of a change.”

The next slide replaced the cylinder with coloured domains containing Java services, APIs, Kafka topics, Redis clusters and several kinds of datastore. Vale explained how the new platform had made ownership explicit, allowed independent releases and isolated failures that had previously stopped branch operations.

Daniel knew the presentation well. He had contributed to it. What troubled him that morning was the contrast between the diagram and the meeting invitation. Twenty-three teams seemed a large audience for a change that the architecture was meant to contain.

“Can we show the address path before the principles?” he asked. “I am still having trouble explaining why this particular change reaches so many teams.”

Vale brought the relevant slide forward. Four domain services required Java changes. Five consumers, an event publisher and the legacy bridge brought the backend total to eleven deployables. There were also an experience API, the web and branch frontends, the mobile application, six Kafka schemas, reporting feeds and changes in several stores and projections.

“Those are components, not eleven separate business decisions,” Vale explained. “Some carry the information between the areas that interpret it. The estimate will separate those costs.”

Mark Bell, a principal architect recently appointed to the Architecture Council, added that not every consumer could adopt a new event at once. “We will need a period in which old and new meanings coexist. Tax and Communications have different release commitments. Removing a service from the diagram wouldn’t remove that problem.”

Daniel agreed. Mark was good at questions like this, and Daniel had relied on him during two difficult migrations. Nevertheless, he still could not give the programme office a simple account of the change.

“Who decides what the new classification means?” Fred asked.

Helen said that Customer Change would coordinate the requirement. Fred clarified that he meant the distinction itself: which facts would establish residence, which would establish correspondence, and what a successful change would allow the rest of the bank to assume.

“Customer Master owns the association,” Daniel said. “Address Validation owns validation metadata. Onboarding collects it, and the consuming domains own their interpretations.”

Fred waited while Daniel considered his own answer. It was accurate, but it did not identify anybody who could settle a disagreement between those interpretations.

Vale returned to the principles slide. Services had single responsibilities, owned their datastores and communicated through explicit contracts. Business logic belonged in application services. Teams and frontends deployed independently. Integration was continuous.

“Are these defaults or requirements?” Peter asked.

“Requirements, with an exception process,” Vale said. “They address failures we have already experienced. We cannot ask every team to invent its own architecture.”

“No,” Peter replied. “But we will need to understand the failures behind them. Otherwise we cannot tell which requirements are helping with the address change.”

Helen explained the attendance figures. Eleven teams expected to change code, schemas or contracts; three more owned migration, assurance or sign-off. The remaining nine needed enough information to confirm that they were unaffected.

“The request called it one field,” she said, “but the distinction behind it is more substantial.”

Fred nodded. “Then let us keep those two things separate as we investigate. Some of the work belongs to the distinction. Some may belong to the route it has to take through the bank. We should find out which is which.”

2. The Backbrief

When Vale finished, Fred asked him to leave the architecture on the screen. He closed the printed presentation and suggested that the reviewers explain their understanding back to the room. Vale could correct it as they went.

Daniel had used backbriefs during incidents, when a misunderstanding could waste the next hour, but rarely after an architecture presentation. The usual sequence was questions, actions and approval. Fred's proposal made it harder for everybody to leave with a different interpretation while believing they had agreed.

"A customer requests a change through one of several channels," Fred began. "The channel calls an experience API, which calls one or more domain services. Those services update their stores and publish events. Other services update further stores. The screen may read from an API or from a projection, and reporting reconstructs what it needs separately. Is that a fair starting point?"

Vale said it was, subject to differences between journeys. Fred drew the path on the whiteboard and asked where the address operation completed.

"Recording and processing aren't the same thing," Mark replied. "We acknowledge acceptance first. Some downstream work is asynchronous, particularly where another organisation is involved."

"That sounds reasonable," Fred said. "What does the acknowledgement promise?"

Aisha Khan, who led the customer-profile frontend team, brought up the screen. Its confirmation said **ADDRESS CHANGE ACCEPTED**. The response came from Customer Master, but the screen did not name that service, nor should it have expected a customer to know what it was.

"It means the request has been recorded," Aisha said. "I can see the wording might imply more. We need to check what customers think has happened."

Peter asked Daniel to count the production services. There were 187 in the catalogue, although the diagram showed only the principal domains. When

Vale explained the reasons for separating them, Peter wrote down information, source code, process, deployment, failure, data and team.

“These are different kinds of separation,” Peter said. “A service can provide several at once. I am asking whether you wanted all of them for every service, or whether some came with the mechanism you selected.”

Mark said that separate ownership and release schedules were often deliberate. “A supplier team can be responsible for a service in a way that is much harder when its code shares a deployment with five other suppliers. A process boundary also makes some kinds of interference difficult.”

Peter accepted the point. “And for the address service, which independence matters most?”

The answer varied between address validation, customer associations and downstream interpretation. Daniel opened the recent programme report. Around sixty per cent of regulatory initiatives touched more than five domains. Those figures did not mean every boundary was wrong, but they did mean co-ordinated change was ordinary work rather than an exceptional inconvenience.

Fred returned to the confirmation message. Ed, who had been listening, wrote a question on a sheet of paper and passed it to Daniel: *If this address change succeeds, what facts become true?*

The room supplied several answers. Customer Master recorded an association. A physical place might be validated. Financial Crime might require rescreening. Tax could reconsider residence. Communications could change delivery. Some projections would lag.

Ed asked Aisha which of those facts the person reading the screen could safely infer. She said that the screen currently gave too little information to distinguish them.

“Then we should identify the promise before discussing whether the operation is complete,” Ed said. “There may be several completion points. There must not be one word pretending to identify all of them.”

Vale approached the board to clarify the path. Mark corrected an event arrow; a tax representative explained that its consumer actually used a derived event. Daniel added the transformation service and the reconciliation path. The clean diagram accumulated annotations until it no longer fitted on the board.

“The original is the target architecture,” Vale said when Fred asked about the difference. “This contains implementation detail.”

“Some of it is detail,” Fred replied. “But the transformation determines what Tax receives, and the reconciliation determines when it can trust the result. We will have to keep enough of the implementation in view to explain the business operation.”

Daniel saved a photograph before the next meeting’s occupants erased the board. He had expected the backbrief to expose gaps in the visitors’ understanding. It had exposed several in his own, including a question he should have been able to answer: what, exactly, had the bank promised the customer?

3. The State That Never Existed

The incident happened the following Monday. At 10:14, a customer changed her declared residential address through the web application. Her previous address had been verified eighteen months earlier. Customer Master recorded the replacement and published an event, which reached Customer Summary in less than a second.

Customer Summary held the most recent residential address it had received and the most recent residential-verification status it had received. It did not record whether those two values referred to the same address. When the new address arrived, the projection combined it with the existing verification status and sent that combination to the branch screen.

At 10:16:33, a branch employee opened the customer's profile. She was completing a periodic customer-information review and had changed her address online while waiting to be seen. The employee saw the new address beside a green **VERIFIED** badge.

"Have you already provided evidence for this address?" he asked.

The customer said she had not. The employee checked the screen again, assumed the bank had evidence from another channel and selected **EVIDENCE ALREADY HELD**. The review completed. Twenty-seven seconds after he had opened the profile, a delayed event reached the projection and changed the badge to **VERIFICATION PENDING**.

The misleading combination had been available for almost three minutes. The employee had seen it for twenty-seven seconds, which was long enough to make a decision the bank would now have to correct.

No message had been lost. The services remained available, the retries succeeded and the distributed trace showed where the delay had occurred. Yet the screen had made a statement that no component had established: the customer's new residential address had been verified.

Daniel had designed Customer Summary. He had approved the subscriptions, the projection and the rule that assembled the latest available customer facts into one document. The design had solved a real problem: the web and

branch teams had previously coordinated six calls to build a slow, inconsistent page. His decision had given them one responsive interface. It had also given that interface the authority to combine facts whose relationship it did not understand.

At the incident review, Mark projected the two payloads. The address event contained a customer identifier, an address identifier, a purpose and a version. The verification event contained a customer identifier, a purpose, a status and a timestamp. It did not identify the address or version to which the verification applied.

“The immediate defect is in the contract,” Mark said. “We need the verified address identified throughout the path. The projection must not attach a status to whichever address happens to be current.”

Aisha asked whether the screen could display the old status as stale. Ed pointed to the screenshot.

“The old status may still be true of the old address,” he said. “It becomes false when attached to this one. An age label would not repair the missing relationship.”

Daniel agreed with Mark’s diagnosis. The identifier would need to survive evidence capture, the verification decision, the published result and every projection that joined it to an address. They could repair the defect within the current architecture, and he wanted that work started immediately.

“Then we should be careful what we infer,” Mark said. “A broken event contract doesn’t establish that the services should be combined. A single application could make the same mistake.”

“Yes,” Daniel replied. “But I don’t think I understood the cost we were accepting. I treated the fields as independently current. They weren’t. We separated them and left each consumer to preserve a relationship we had barely described.”

He wrote the relationship on the board: **EVIDENCE E SUPPORTS VERIFICATION OF ADDRESS A AT VERSION V**. The verification badge represented the result of applying a policy to that relationship. It could not safely be reduced to a customer’s latest status.

Mark explained what eventual consistency promised here. Once writes stopped, messages were delivered and consumers recovered, the projection

would converge on the authoritative values. That did not establish which intermediate combinations were meaningful.

“That is the distinction we need,” Ed said. “The system may take time to learn something. During that time it still needs rules about what it may assert.”

Margaret Shaw joined remotely from the Atlas support area. She was sixty-eight, wore bright red glasses and worked three days a week under a contract renewed every six months. She shared the old schema, in which evidence referred to an identified address record and verification referred to the evidence. Replacing an address closed its previous period without transferring its verification to the replacement.

“Would Atlas have prevented this?” Aisha asked.

“The representation would have made it harder,” Margaret said. “A programmer could still join the wrong rows or put the wrong badge on a screen. What the database gave us was a relationship we could preserve and test. It didn’t guarantee that every reader would use it correctly.”

Mark examined the keys. “We can preserve that relationship across services too.”

“Certainly,” Margaret replied. “Then you need to carry it through the separation. You have to identify what was verified, preserve that identity and decide what a consumer does when only half the answer has arrived. The cost is real, even when it is worth paying.”

Fred had been reading Daniel’s original architecture decision. He asked why it specified the most recently available customer attributes without defining valid combinations.

“We reviewed freshness and availability,” Daniel said. “We checked event ordering within each stream. We didn’t ask whether values from different streams could form a sentence that nobody had made true.”

Ruth Ellis had not yet joined the programme, so Aisha offered to arrange observations with branch staff. They needed to know how many other labels carried assumptions that the implementation did not justify. Helen asked Operations to identify affected reviews and ensure that correcting the projection did not leave the decisions made from it unexamined.

The meeting ended with a repair plan rather than an architectural verdict. Daniel stayed behind to copy the evidence identifiers from the board. The

branch employee had seen one address and one green badge. He had not been asked to reason about projections or delayed events, and he should not have had to.

The interface had turned two values into a sentence. The sentence was false.

4. The Horror Exhibit

Atlas had begun in the late 1990s as a customer database for the branch network. It had since absorbed online servicing, regulatory classifications, product relationships, tax reporting, document preferences and several kinds of information that no longer matched the bank's organisation.

It ran on Oracle, with one principal application schema containing 1,842 tables, more than 6,000 stored procedures, hundreds of views and a collection of jobs that everyone agreed should be better documented. Prefixes such as CUST_, TAX_ and COMM_ suggested ownership. Permissions enforced much less of it. Some applications used procedures as interfaces; others wrote directly to the tables whenever a deadline made the interface inconvenient.

The modernisation programme introduced this history to new employees in a presentation called **WHY MERIDIAN**. Daniel had delivered it himself. Taking the visitors to Margaret's desk, he realised he had spent more time explaining why Atlas should go than learning what would be lost when it did.

Vale had described Margaret as a key-person dependency. She described herself as the person they called after the key people left. Fred asked about her contract, and she explained the six-month renewal cycle while opening the branch address-change procedure.

"They review whether I'm still a risk," she said. "The work takes rather longer to review."

The procedure was 412 lines long. It validated input, checked customer status, applied jurisdiction rules, changed address relationships, recalculated flags, queued reviews, wrote history and prepared an extract. Some of those procedures called others. Daniel watched the call tree expand and explained that discovering the actual operation required following several levels of indirection.

"Yes," Margaret said. "We did not make it easy. Some of the names are misleading, and some branches survive only because nobody has been confident enough to remove them."

Mark stopped at a condition involving trusts, British addresses and a date in 2014. He asked whether it still represented a business requirement or merely compatibility with an old feed.

“That one protects historical classifications,” Margaret said. “It prevents a later rule from rewriting what we were entitled to conclude at the time. There are two nearby that exist for an extract we retired. Those can probably go.”

She did not defend the procedure as a whole. It held too much knowledge, some of it in the wrong place. What she defended was the need to determine why a branch existed before moving or deleting it.

Fred asked what happened when the operation succeeded. Margaret described one commit covering the customer change, address history, evidence relationships, audit record and queue entry. If a required part failed, the transaction rolled back. Work performed by the queue was a later operation; the commit promised that it had been durably requested, not completed.

“And which rules survive somebody bypassing the procedure?” Ed asked.

Margaret showed foreign keys and simple checks, a function-based unique index for the permitted current primary address, and the locking protocol used before checking historical overlap. A trigger attempted to catch direct writes, although those paths required particularly careful testing because they did not all enter through the same procedure.

“That is one reason we moved to service ownership,” Mark said. “We wanted one supported write path instead of discovering another batch account after each incident.”

“It was a good reason,” Margaret replied. “The question is whether the supported path also preserves the rules when two callers arrive together. Moving it into Java doesn’t answer that by itself.”

Mark raised deployment rather than location. A database change affected persistent state and could make an older application incompatible during a rolling release. Reverting a Java binary did not reverse a data migration, and suppliers had sometimes discovered those differences in production.

“I agree,” Margaret said. “We keep the SQL in source control, but that only versions the instructions. It doesn’t make the migration reversible. We have to introduce compatible forms, move the data, change the readers and remove the old form when nobody needs it. Some changes are much harder than others.”

Daniel said that the same compatibility problem existed across event versions. Mark agreed, but pointed out that it had different operational consequences. They would need to compare those consequences for the proposed change, rather than treating the ability to store a script in Git as an answer to deployment safety.

Fred asked which parts of the address implementation changed most frequently. The framework and channel interfaces changed regularly. Evidence policy varied by jurisdiction. The rule preventing two exclusive current residential relationships had remained stable, although the conditions under which exclusivity applied needed care.

“We should distinguish those kinds of change,” Fred said. “A policy that varies, an invariant you intend to preserve and a framework you expect to replace do not necessarily need the same design.”

Peter had been examining the schema browser. He asked whether Customer, Tax, Screening and Communications could each publish a controlled interface while keeping their own tables private. Margaret said Oracle could enforce more separation than Atlas used. Separate schemas, views, routines, roles and grants were all available; Atlas’s history had left most of the decisions in names and custom instead.

“Separate databases make casual interference harder,” Mark said. “Somebody cannot add a join simply because the tables happen to be there.”

“They do,” Margaret replied. “But Atlas’s broad grants don’t establish that every shared engine must have broad grants. We should compare the ownership mechanisms we could use, not assume the only alternative to Meridian is this exact mess.”

Fred asked which repair accounts could bypass the normal path. Ed asked how concurrent changes to two purposes acquired locks. Peter asked what the historical dates meant. The discussion moved away from whether stored procedures were modern and towards what the system actually guaranteed.

As they left, Margaret asked Peter whether he had given his surname. He said he had not. She studied him for a moment, then returned to the screen without asking again. Daniel noticed, and was relieved that somebody else had noticed too.

5. How Large Is Meridian?

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theparnassuite>.

6. The Lost Context

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theparnassuite>.

7. The Paper

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theparnassuite>.

8. The Cost

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theparnassuite>.

9. What Is a Module?

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theparnassuite>.

10. Two Changes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theparnassuite>.

11. Margaret's Theory

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theparnassuite>.

12. The Third Design

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theparnassuite>.

13. The Module Promise

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theparnassuite>.

14. Where Meridian Was Right

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theParnassuite>.

15. The Screen

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theparnassuite>.

16. The Read Side

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theparnassuite>.

17. The Second Dashboard

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theparnassuite>.

18. The Second System

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theparnassuite>.

19. The Names

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theparnassuite>.

20. The Reviews

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theparnassuite>.

21. What Happened Next

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theparnassuite>.

Author's Note

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/theParnassuite>.

Buy the book!

What you read was just a sample. Why not buy the full book?