Joyful Health

The Denial Architecture Model

How Healthcare Can Design Control into Revenue

Table of contents

The Denial Architecture Model 01

Executive Summary 02

The $125 Billion Question 05

The Infrastructure Gap in Healthcare Revenue 08

Introducing the Denial Intelligence Stack 11

The Denial Management Operating System 23

Denials as Financial Signals 26

The CFO Translation Layer 28

The Denial Architecture Maturity Model 30

Prevention as a Designed Control System 33

Diagnostic Self-Assessment 36

Designing Control into Revenue 38

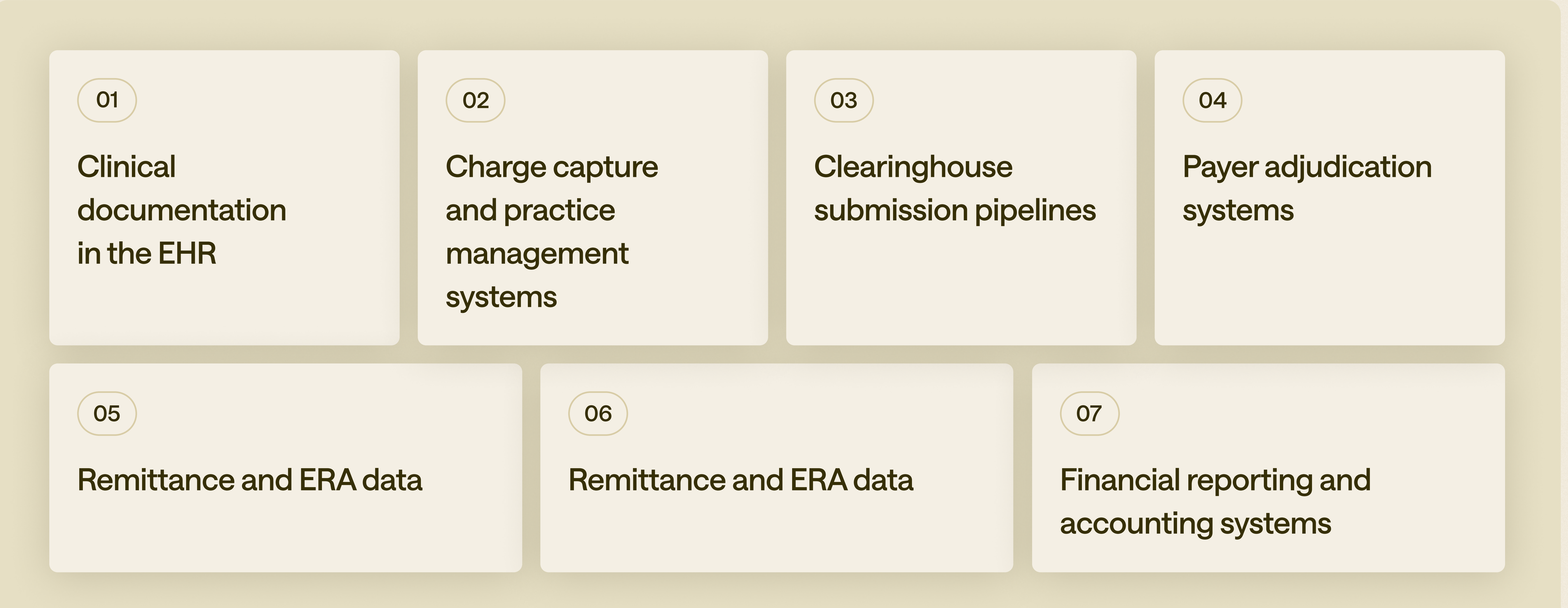
About Joyful Health 41

Executive Summary

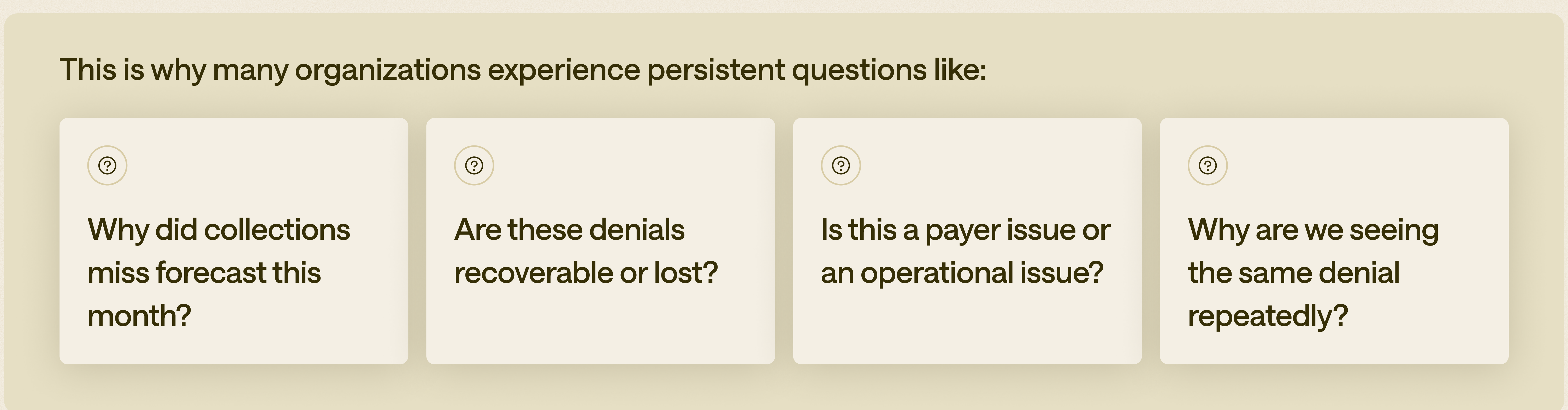
Executive Summary

Healthcare organizations are not short on revenue data, they are short on visibility into what actually happened to their revenue.

Between the moment a patient receives care and the moment a payment reaches the bank account, revenue signals pass through a complex network of systems.



> Each system captures a portion of the story.
> None capture the full narrative.
>
> When revenue volatility appears: denial spikes, delayed payments, unexplained forecast variance, and leaders often lack the structural visibility required to explain what happened.



The underlying issue is rarely staffing alone. It is **architecture.**

Most healthcare organizations lack a system that converts operational denial signals into structured financial intelligence.

This paper introduces the **Denial Architecture Model**, a framework for transforming denial events into:

- Operational insight
- Financial predictability
- Engineered prevention.

The model introduces a structured intelligence stack, an operational feedback system, and a maturity model for revenue control.

When denial architecture is implemented effectively, revenue operations shift from reactive recovery to proactive, **designed financial control**.

Joyful Health

Becky Carlson

Head of RCM, Joyful Health

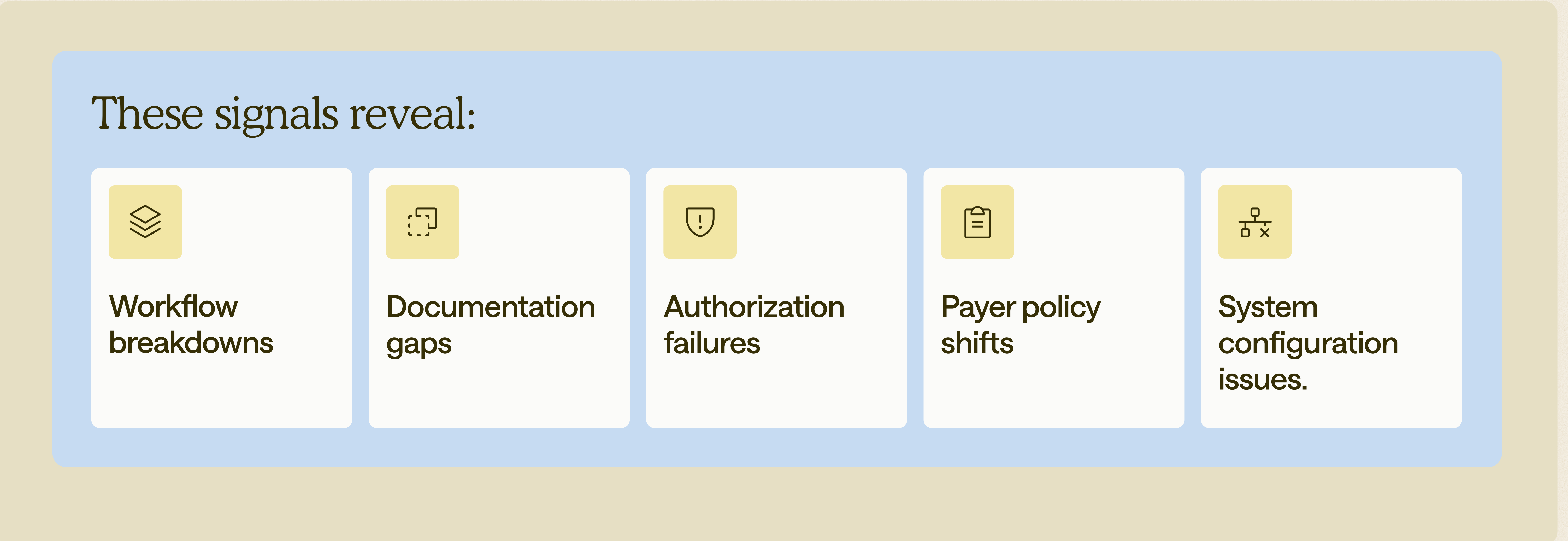
02

The $125 Billion Question

The $125 Billion Question

Estimates suggest that U.S. healthcare providers lose or fail to collect more than $125 billion annually due to denied or underpaid claims, yet the magnitude of the problem often obscures its real cause.

Denials are rarely just billing errors, they are signals generated by a complex payment system.



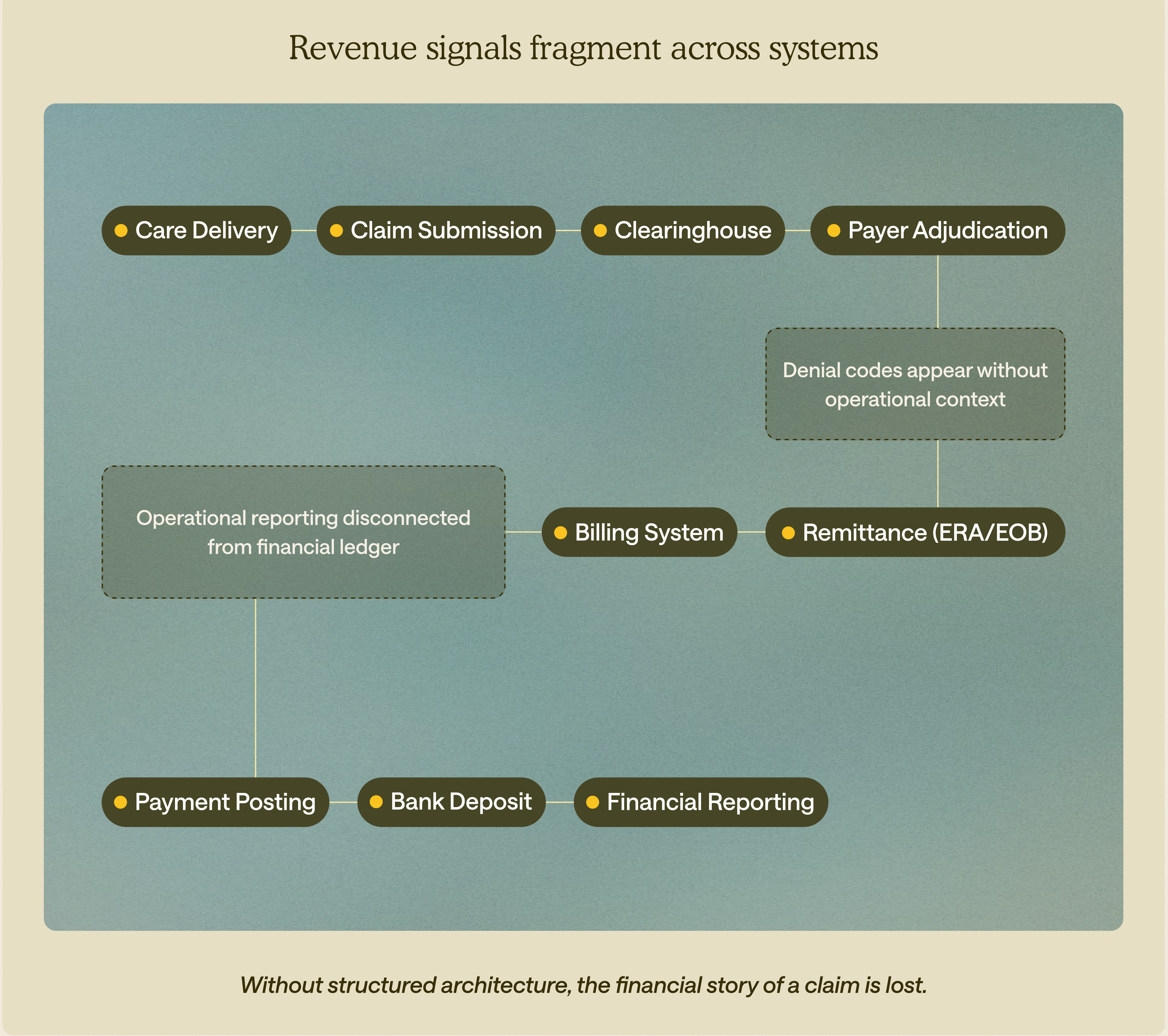
But because denial information is fragmented across systems, organizations often see the symptoms of revenue disruption without seeing the mechanism behind it.

Finance teams observe

- Unexpected collection slowdowns
- Increasing A/R aging
- Fluctuating net collection rates
- Rising appeal volumes.

Revenue cycle teams see

- Growing denial queues
- Payer inconsistencies
- Repeated denial categories



Without structured architecture, the financial story of a claim is lost.

03

The Infrastructure Gap in Healthcare Revenue

The Infrastructure Gap in Healthcare Revenue

Most industries operate with a financial signal architecture:

- Payment companies rely on platforms like Stripe.
- Financial institutions rely on data infrastructure like Plaid.
- Retail organizations track revenue through tightly integrated commerce systems.

These systems convert operational activity into **structured financial visibility**. Healthcare revenue does not operate this way.

Instead, revenue signals move through a fragmented network of systems that were never designed to function as a unified financial architecture.

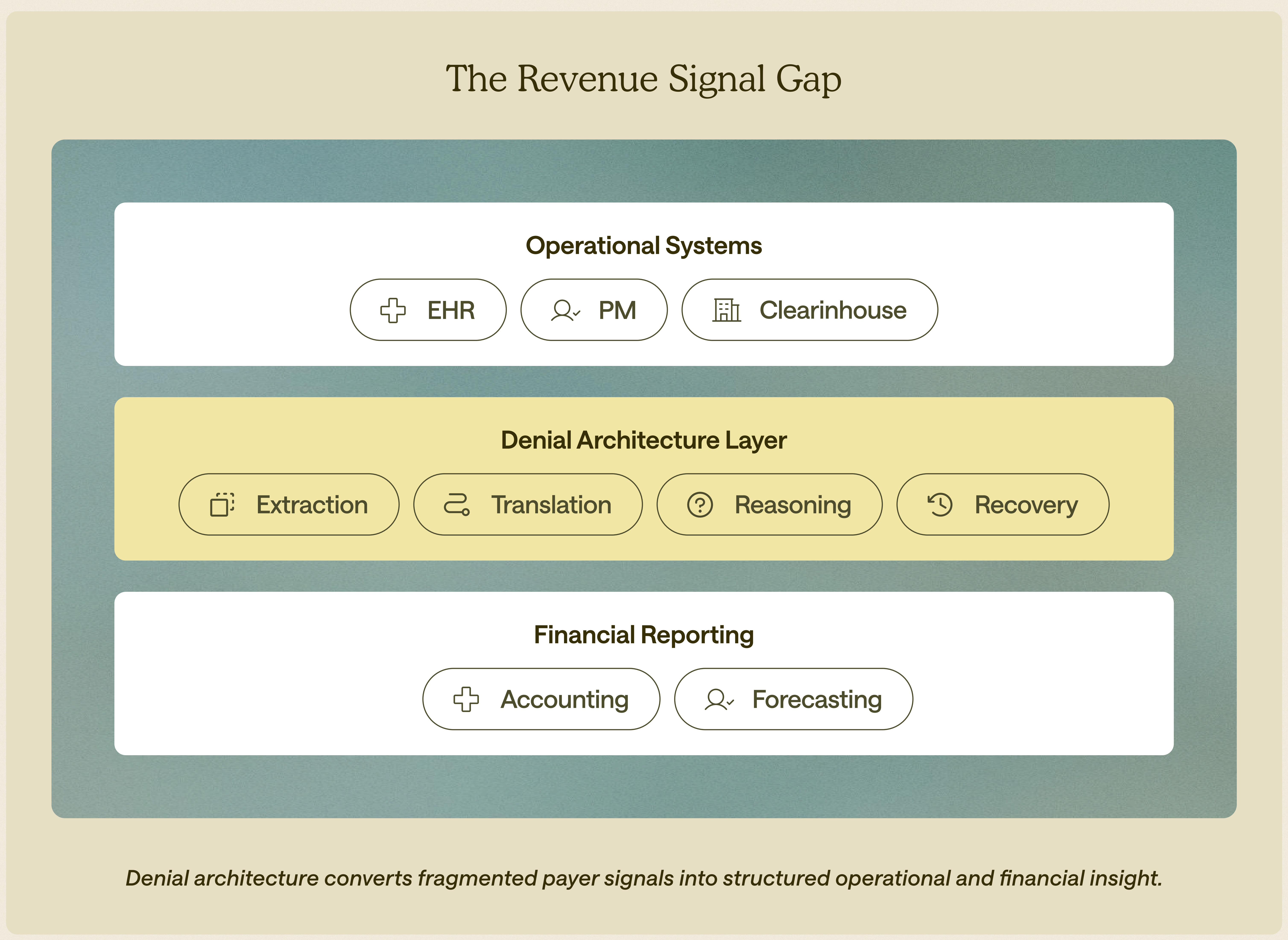
Consider the typical path of a healthcare claim:

1. Care delivery
2. Claim submission
3. Clearinghouse transmission
4. Payer adjudication
5. Remittance advice
6. Payment posting
7. Bank deposit
8. Financial reporting

As a result, organizations often discover discrepancies between operational revenue reports and actual bank deposits.

Without a structured architecture connecting these signals, leaders must rely on lagging financial metrics to understand revenue performance.

Denials appear as operational noise. In reality, they are **the most visible signal of where revenue architecture is failing**.



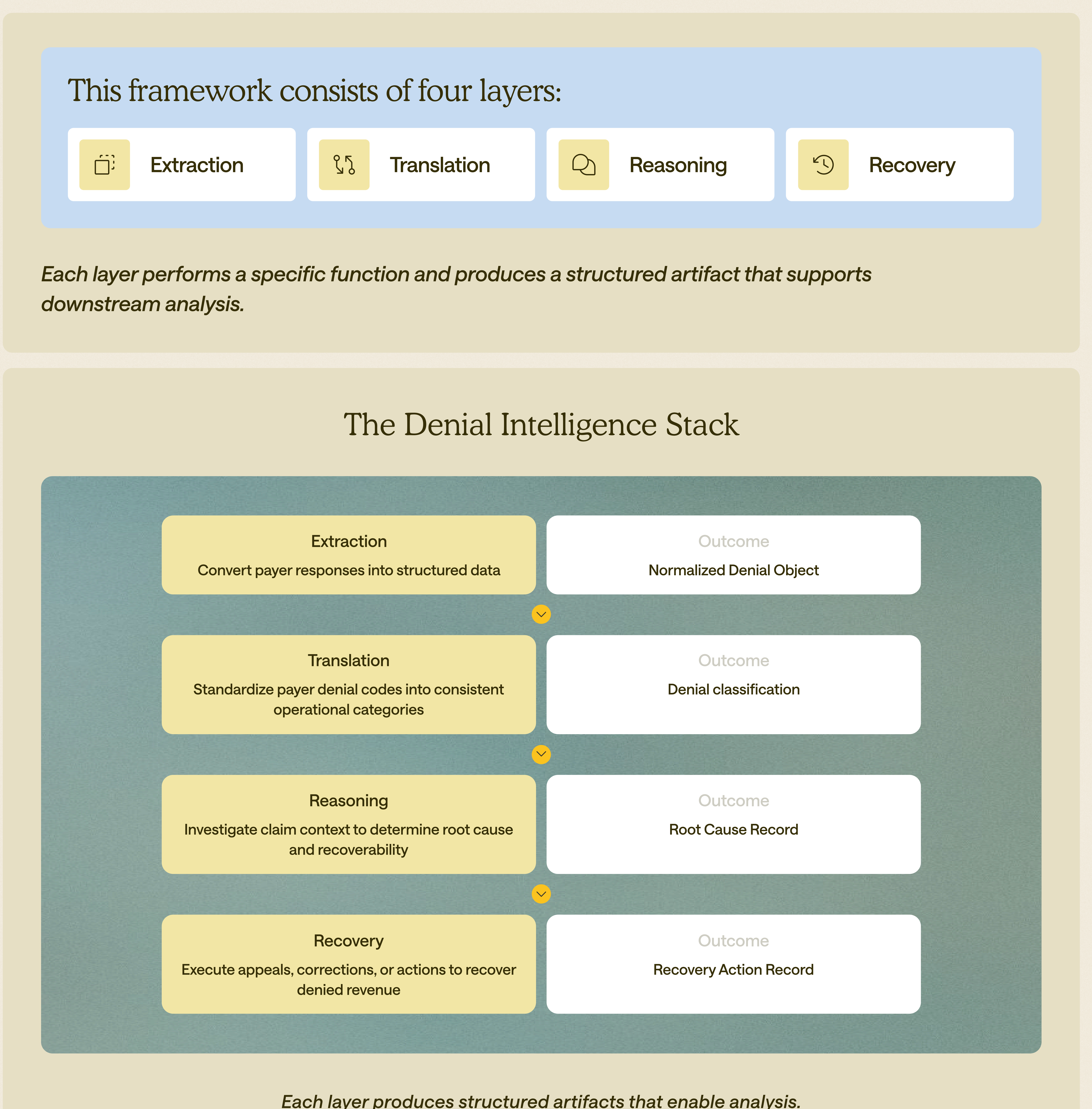
Denial architecture converts fragmented payer signals into structured operational and financial insight.

04

Introducing the Denial Intelligence Stack

Introducing the Denial Intelligence Stack

The Denial Architecture Model begins with a framework that converts payer responses into operational decisions.



Extraction

Extraction converts payer responses into structured data. **This includes information from:**

- Electronic remittance advice (ERA)
- Explanation of benefits (EOB)
- Adjustment codes
- Claim identifiers
- Service line data
- Claim submission history

Extraction is purely mechanical. Its responsibility is to ensure that denial signals exist in a structured format. It does **not interpret meaning or determine whether the denial was correct, preventable, or recoverable. It simply ensures the signal exists in a usable format.**

The output of this layer is a **normalized denial object**, a structured representation of the denial event.

What This Means in Practice

Extraction is not defined by how sophisticated the tooling is. It is defined by whether all relevant information is brought together into one place in a consistent and usable format.

How Extraction Shows Up Operationally

01 **Technical Extraction**

In more mature environments, extraction is performed through automated data pipelines. This may include:

- Pulling 835 ERA files via SFTP
- Ingesting 837 claim submissions via API
- Parsing clearinghouse responses programmatically
- Loading data into a centralized warehouse or reporting layer

The outcome is a continuously updated, structured dataset of denial-related information.

02

Operational / Manual Extraction

In many organizations, extraction is performed through reporting and manual consolidation. This may include:

- Running denial reports from the EHR
- Exporting clearinghouse reports
- Pulling ERA summaries
- Combining data into Excel, Google Sheets, or a BI tool

The outcome is a working dataset that brings together denial-related fields into a single view.

What Teams Are Actually Doing

Operationally, extraction consists of:

- Pulling data from multiple systems
- Exporting or ingesting reports
- Aligning and standardizing columns across sources
- Ensuring key fields are present, such as:
    - Claim ID
    - Payer
    - CPT/HCPCS and diagnosis codes
    - CARC and RARC codes
    - Billed, allowed, and paid amounts
- Creating a single dataset where each row represents a claim or service line

This dataset represents the consolidated view of denial activity.

Example

A billing team exports a denial report from their EHR, pulls ERA data from their clearinghouse, and combines both into a spreadsheet. They align the columns, ensure denial codes are included, and create a single dataset where each row represents a denied claim.

Translation

Payer denial codes are often ambiguous and inconsistent across payers. Translation standardizes this information. The same issue can appear under different codes depending on the payer, and many codes provide very little usable explanation on their own.

This layer maps payer codes into a structured denial grouping so that denials can be categorized consistently regardless of payer. The output is a **denial classification record**.

Importantly, translation does not determine the root cause of the denial. It only standardizes the category.

What This Means in Practice

Translation is not about determining why a denial occurred. It is about ensuring that similar denial signals are grouped together in a consistent way.

At its core, translation answers the question:
"What type of denial is this?" — not "what caused it?"

How Translation Shows Up Operationally

01 **Automated Translation (Rules-Based Classification)**

In more mature environments, translation is handled through logic applied to denial codes.
This may include:

- Mapping CARC and RARC codes to predefined denial categories
- Applying rules within billing systems, clearinghouses, or data pipelines
- Using lookup tables or logic layers to standardize classifications

Examples:

- CO16 → "Missing or Invalid Information"
- CO197 → "Prior Authorization Required"
- CO29 → "Timely Filing"

Multiple codes may map to the same category, depending on how the organization defines its grouping logic.
The outcome is a consistent, system-driven classification applied at scale.

02

Operational / Manual Translation

In many organizations, translation is performed manually as part of reporting or review workflows. This may include:

- Reviewing denial codes within a report or spreadsheet
- Assigning a category column based on interpretation of codes
- Grouping similar denials together for tracking and reporting

For example:

- A team reviews a list of CO16 denials and assigns them to "Missing Information"
- Multiple related codes are manually grouped into a single category within a report

The outcome is a categorized dataset that enables basic aggregation and visibility.

What Teams Are Actually Doing

Operationally, translation consists of:

- Reviewing denial codes (CARC/RARC)
- Mapping those codes to internal categories
- Grouping similar denials under a shared label
- Creating a new field (e.g., "Denial Category") within the dataset

This results in a structured classification layer on top of the extracted data.

Example

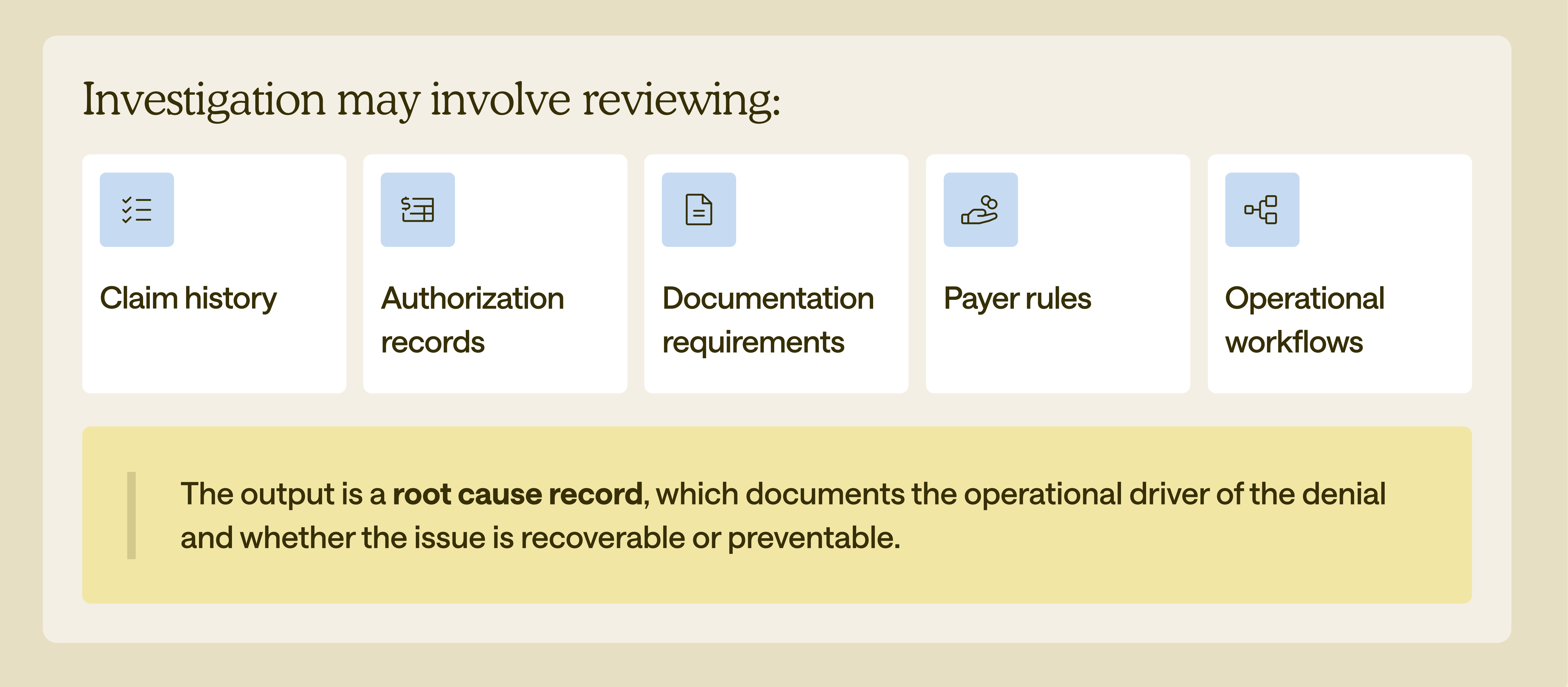
A team reviews denial data and maps multiple codes (e.g., CO16, MA130) into a category labeled "Missing or Invalid Information." Similarly, codes related to authorization (e.g., CO197) are grouped under "Prior Authorization."

This categorization is added as a column in the dataset, allowing the team to aggregate and report on denials by category rather than by individual code.
This is translation in practice.

Joyful Health

Reasoning

Reasoning is the investigative layer. This is where organizations determine **why a denial occurred**.



What This Means in Practice

Reasoning moves beyond categorization and into interpretation.
While translation answers "What type of denial is this?", reasoning answers:
"Why did this denial actually happen in this specific instance?"

This requires synthesizing multiple data points and applying contextual understanding of payer behavior and operational processes.

How Reasoning Shows Up Operationally

01 **Limited Automation (Advanced / Vendor-Supported)**

Reasoning is difficult to fully automate due to its reliance on context and judgment.
In more advanced environments, organizations may:

- Use rules engines or decision frameworks to suggest likely root causes
- Leverage vendor solutions (e.g., platforms like Joyful Health) to assist in pattern recognition and investigation
- Apply structured workflows to guide analysis

However, even in these environments, human validation is often required.

02

Operational / Manual Investigation (Most Common)

In most organizations, reasoning is performed manually by billing or RCM teams. This typically includes:

- Reviewing claim history and prior submissions
- Checking prior authorization status and requirements
- Validating documentation and medical necessity
- Referencing payer policies and guidelines
- Evaluating provider enrollment and network status
- Comparing denial codes against known payer behaviors

The outcome is a determination of the underlying cause of the denial.

What Teams Are Actually Doing

Operationally, reasoning consists of:

- Using the extracted and translated data as a starting point
- Investigating additional context across systems and documentation
- Identifying inconsistencies between the denial and payer rules
- Forming a hypothesis about the true driver of the denial
- Confirming that hypothesis through validation steps

This process results in assigning a root cause to the denial.

Example

A denial is categorized as "Prior Authorization Required" based on its code (e.g., CO197).

However, during investigation:

- The payer's policy indicates that prior authorization is not required for the service
- The team confirms that no prior authorization could have been obtained
- The team evaluates alternative explanations:
  - Prior authorization used as a proxy for medical necessity review
  - Prepayment review requirements
  - Provider network status issues

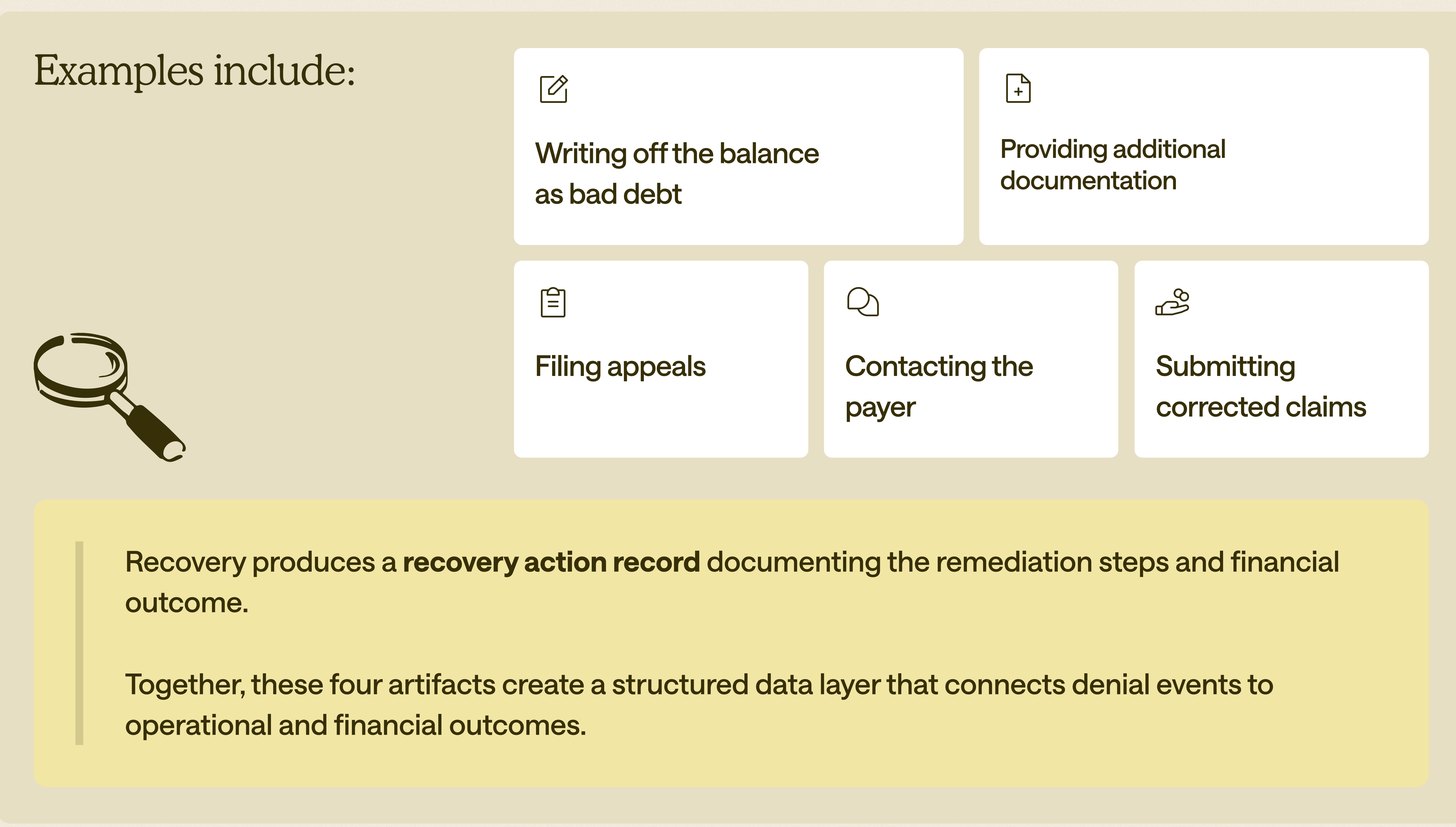
Further investigation reveals:

- If the provider is in-network, the denial is likely related to prepayment or medical necessity review
- If the provider is out-of-network, the denial may be driven by network status rather than true authorization requirements

The root cause is then documented accordingly.
This is reasoning in practice.

Recovery

Recovery is the execution layer. Here the organization performs the financial action required to resolve the denial.



Examples include:

- Writing off the balance as bad debt
- Providing additional documentation
- Filing appeals
- Contacting the payer
- Submitting corrected claims

> Recovery produces a **recovery action record** documenting the remediation steps and financial outcome.
>
> Together, these four artifacts create a structured data layer that connects denial events to operational and financial outcomes.

What This Means in Practice

Recovery is where analysis turns into action.

While reasoning identifies why a denial occurred, recovery answers:
"What are we going to do about it?"

This is the point at which teams engage directly with payers, update claims, and attempt to resolve or close out the denial.

How Recovery Shows Up Operationally

01 Standard Recovery Actions

In most organizations, recovery consists of a defined set of actions, including:

- Submitting corrected or rebilled claims
- Filing formal appeals
- Sending additional documentation or medical records
- Contacting the payer for clarification or status updates
- Escalating issues through payer channels
- Writing off balances as contractual or bad debt when appropriate

These actions are typically performed within the EHR, clearinghouse, payer portals, or via phone and fax.

---

02 Workflow-Driven Execution

Recovery is often managed through operational workflows, such as:

- Work queues assigned to billing or AR teams
- Task-based follow-up schedules
- Appeal tracking logs
- Timely filing and appeal deadline management

The outcome is a structured process for resolving denials at scale.

What Teams Are Actually Doing

Operationally, recovery consists of:

- Reviewing the root cause identified during reasoning
- Selecting the appropriate resolution path (appeal, correction, write-off, etc.)
- Executing the required action within the appropriate system
- Following up with payers as needed
- Tracking progress through resolution

Each action taken is tied back to the original denial.

Example

A denial is identified as a documentation-related issue. The team:

- Gathers the required medical records
- Submits an appeal with supporting documentation
- Follows up with the payer to confirm receipt and status
- Receives a revised determination

If approved, the claim is reprocessed and paid. If denied, the team may escalate or determine that the balance should be written off.

This sequence of actions represents recovery in practice.

Joyful Health

05

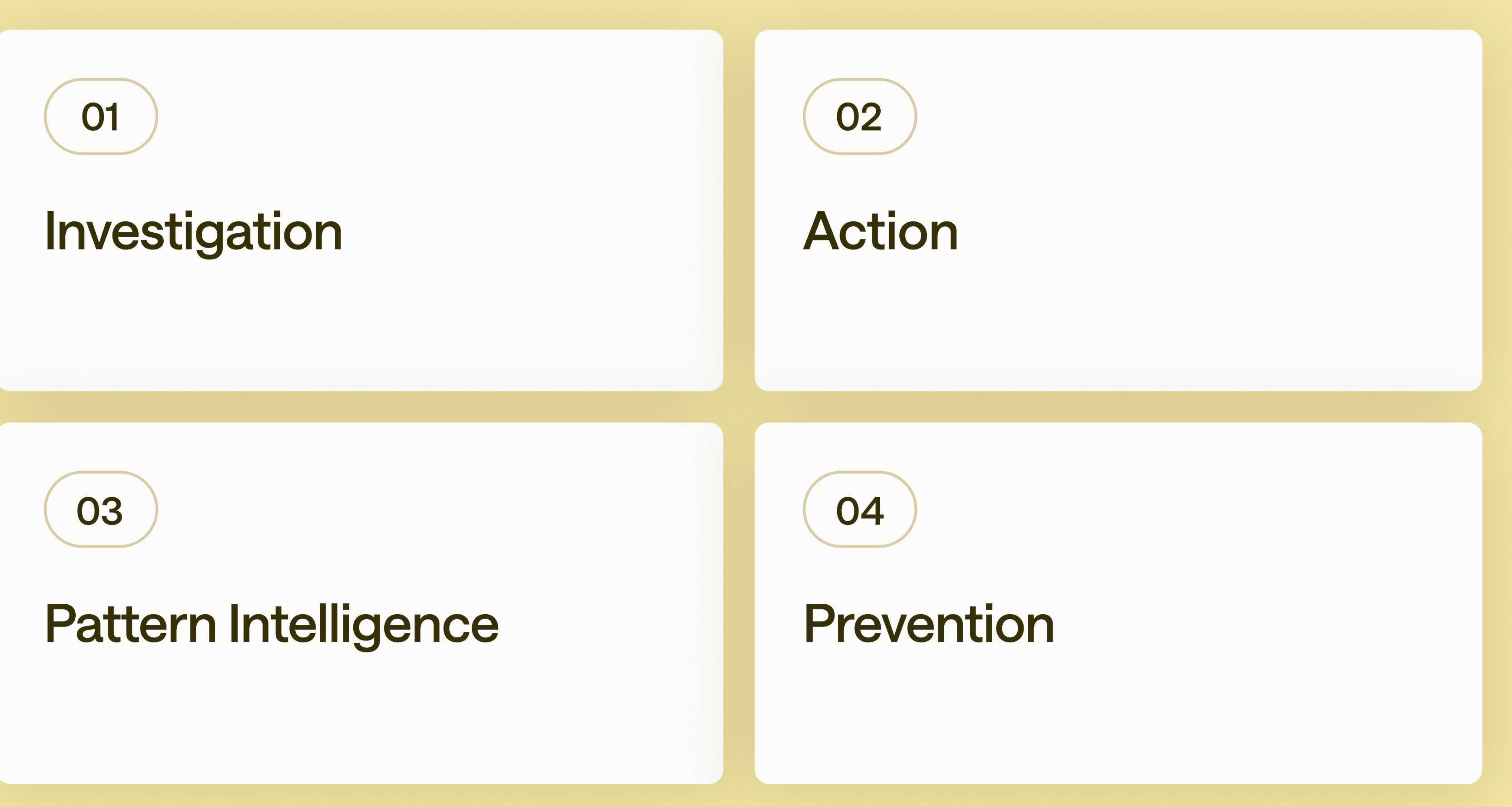
The Denial Management Operating System

Joyful Health

The Denial Management Operating System

While the intelligence stack processes denial signals, denial architecture also requires an operational control system.

This system operates across four structural layers:



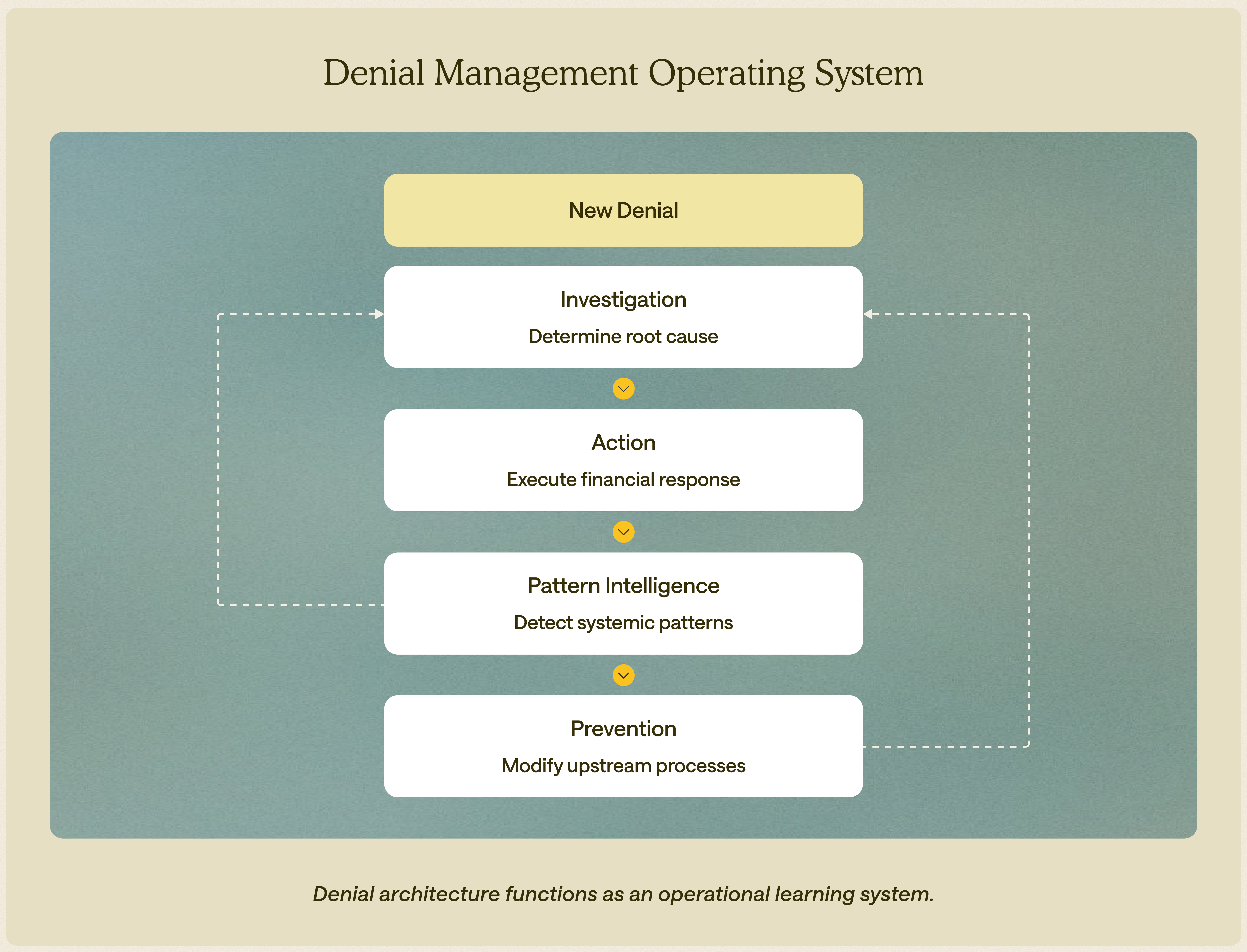
Each layer operates on a different time horizon:

- Investigation and Action operate at the claim level, typically in real time or near real time.
- Pattern intelligence operates at a portfolio level, analyzing trends across many claims.
- Prevention operates on a forward-looking horizon, implementing changes that reduce recurrence.

Signals flow between these layers through structured feedback loops:

- Investigation produces root cause records.
- Recovery produces financial outcomes.
- Pattern intelligence aggregates these signals to identify systemic patterns.
- Prevention uses these insights to modify workflows and reduce future denial occurrence.

When these feedback loops are functioning properly, denial architecture becomes a **continuous learning system**.



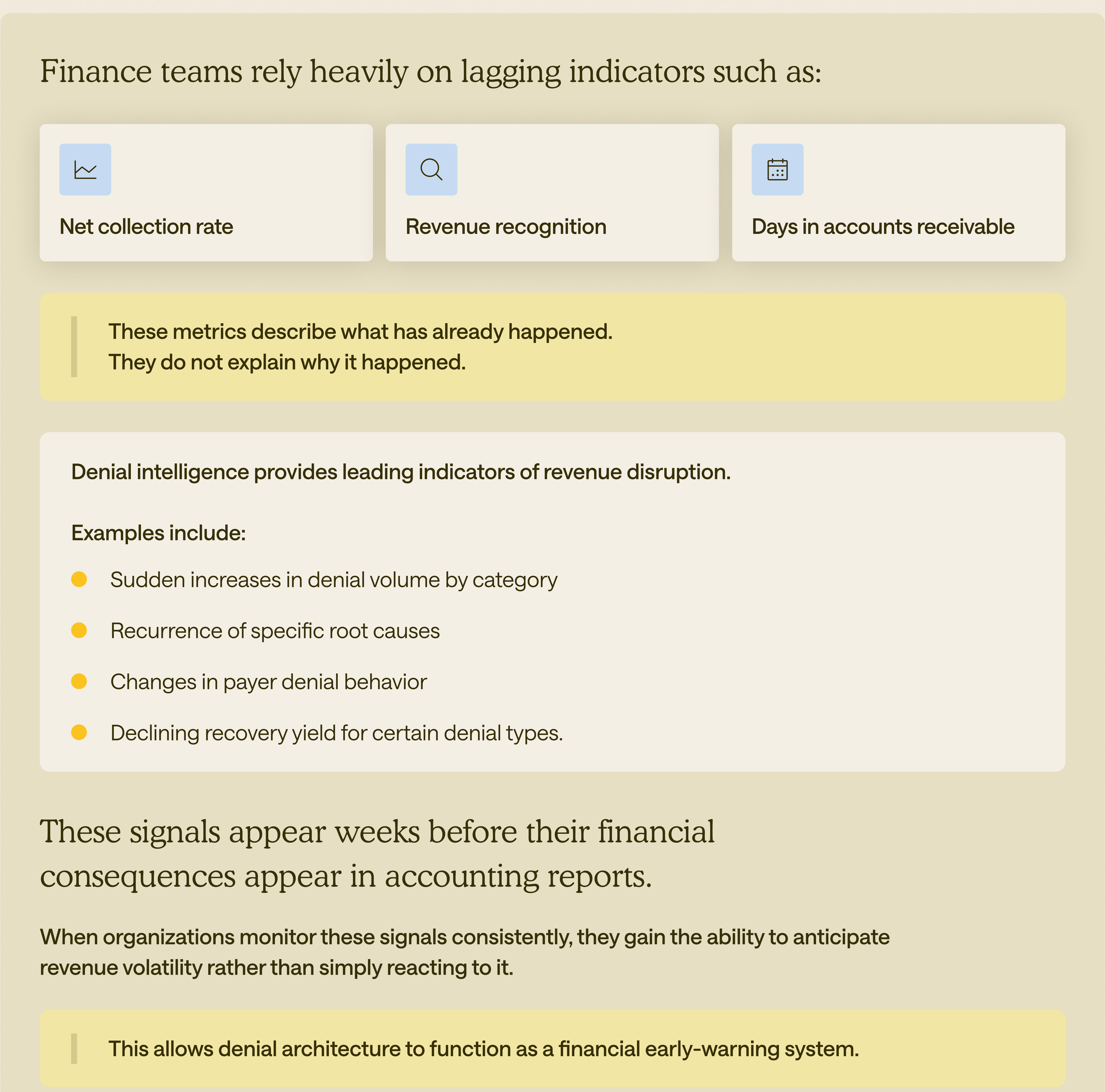
Denial architecture functions as an operational learning system.

06

Denials as Financial Signals

Denials as Financial Signals

Denial architecture becomes most powerful when its outputs translate into financial insight.



Finance teams rely heavily on lagging indicators such as:

- Net collection rate
- Revenue recognition
- Days in accounts receivable

> These metrics describe what has already happened.
> They do not explain why it happened.

Denial intelligence provides leading indicators of revenue disruption.

Examples include:

- Sudden increases in denial volume by category
- Recurrence of specific root causes
- Changes in payer denial behavior
- Declining recovery yield for certain denial types.

These signals appear weeks before their financial consequences appear in accounting reports.

When organizations monitor these signals consistently, they gain the ability to anticipate revenue volatility rather than simply reacting to it.

> This allows denial architecture to function as a financial early-warning system.

07

The CFO Translation Layer

The CFO Translation Layer

Operational denial signals must ultimately translate into financial context.

Root cause records and recovery artifacts influence key financial metrics such as:

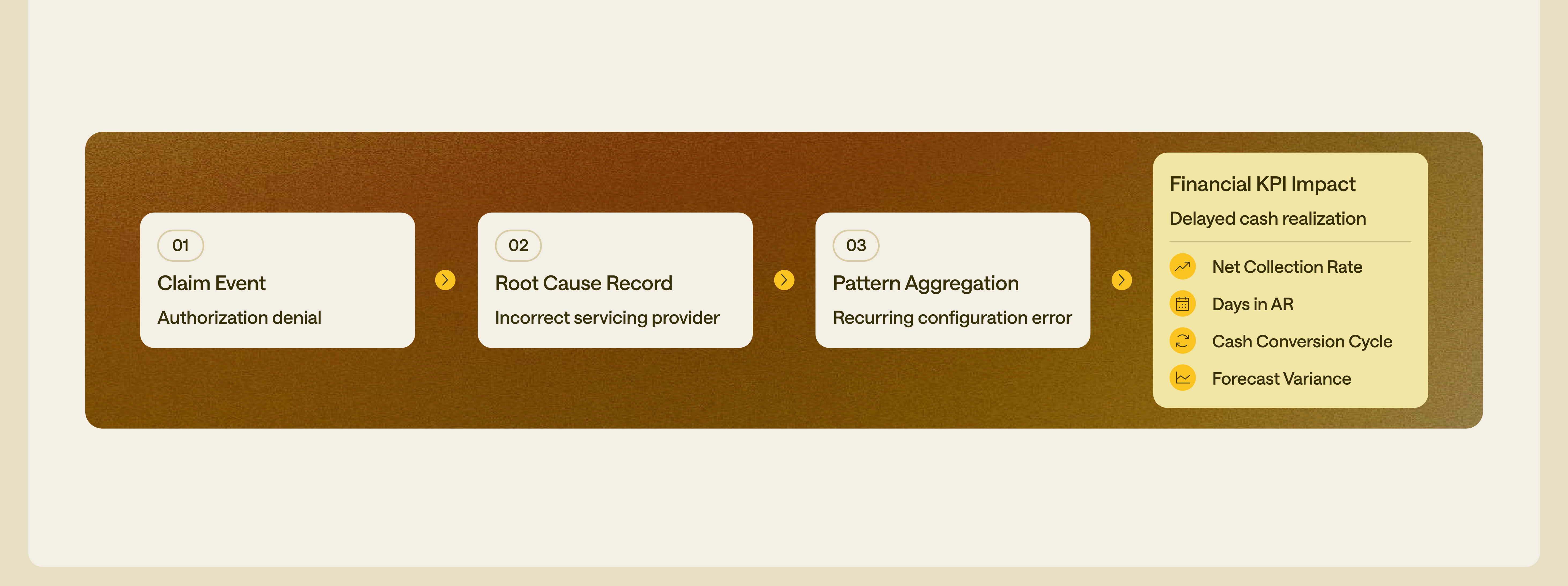
- Net collection rate
- Cash conversion cycle
- Days in A/R
- Cost to collect

Finance teams can incorporate this signal into short-term cash forecasting.

Similarly, a recurring denial pattern related to documentation gaps may indicate systemic operational risk that could affect future revenue realization.

By linking operational denial data to financial outcomes, organizations create a bridge between revenue cycle operations and financial leadership.

This bridge allows finance teams to explain revenue variance using structured operational insight rather than retrospective guesswork.



The Denial Architecture Maturity Model

Joyful Health

The Denial Architecture Maturity Model

Organizations evolve through stages of denial architecture maturity.

Level 1

Reactive Recovery

At this stage, denials are handled through work queues and individual appeals.

Root cause documentation is inconsistent.

Denial data is rarely structured.

> Many organizations remain at this level regardless of size.

Level 2

Structured Investigation

Organizations introduce consistent denial categorization and root cause documentation.

Denial investigation becomes part of the operational workflow.

Level 3

Pattern Intelligence

Root cause data is aggregated to identify systemic denial drivers and payer behavior patterns.

Organizations begin moving from reactive recovery to proactive insight.

Level 4

Prevention Architecture

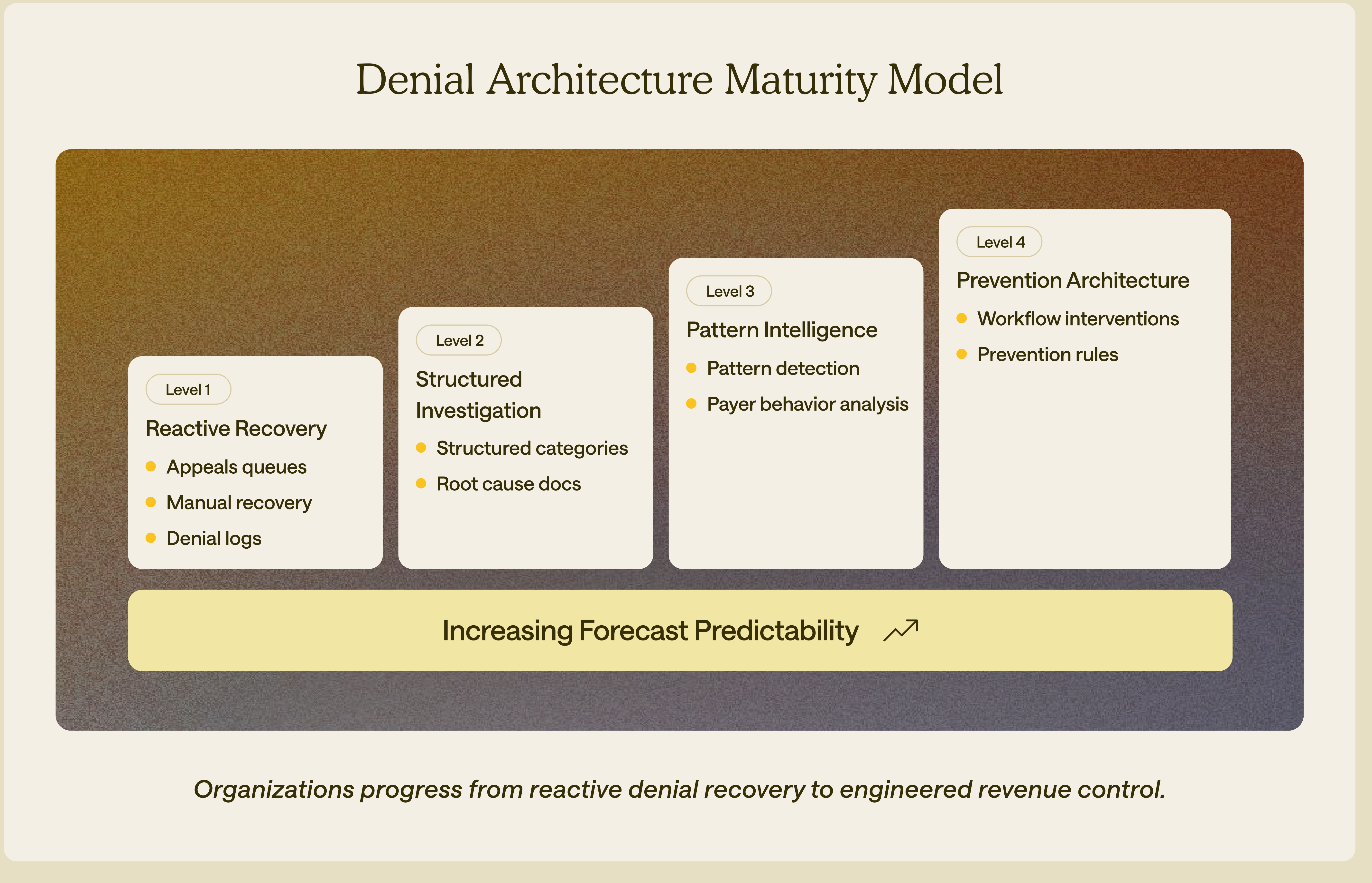
Workflow adjustments and validation rules are implemented to reduce recurrence.

Denial intelligence informs upstream operational changes.

> At this stage, denial architecture becomes a structural component of revenue governance.

Organizations operating at higher maturity levels often experience measurable improvements in financial predictability and forecast reliability.

Denial Architecture Maturity Model



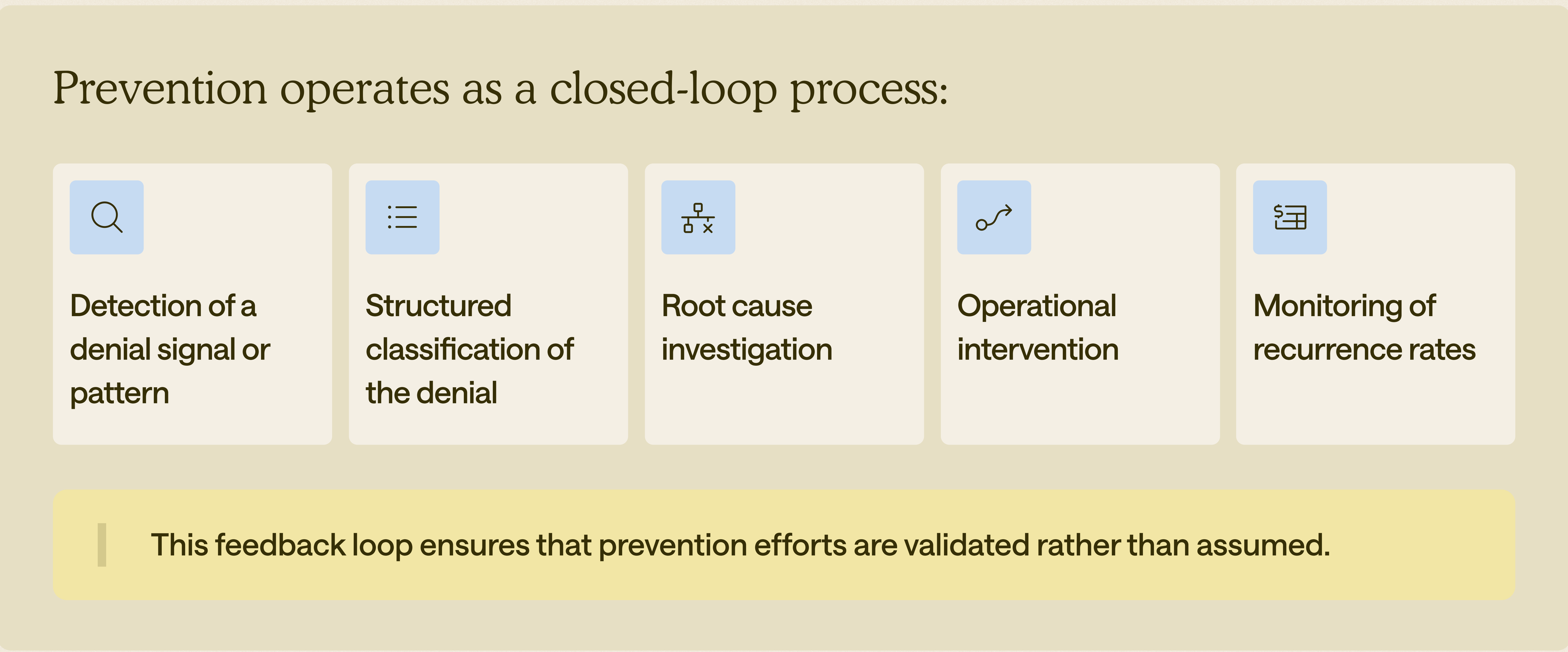
Organizations progress from reactive denial recovery to engineered revenue control.

09

Prevention as a Designed Control System

Prevention as a Designed Control System

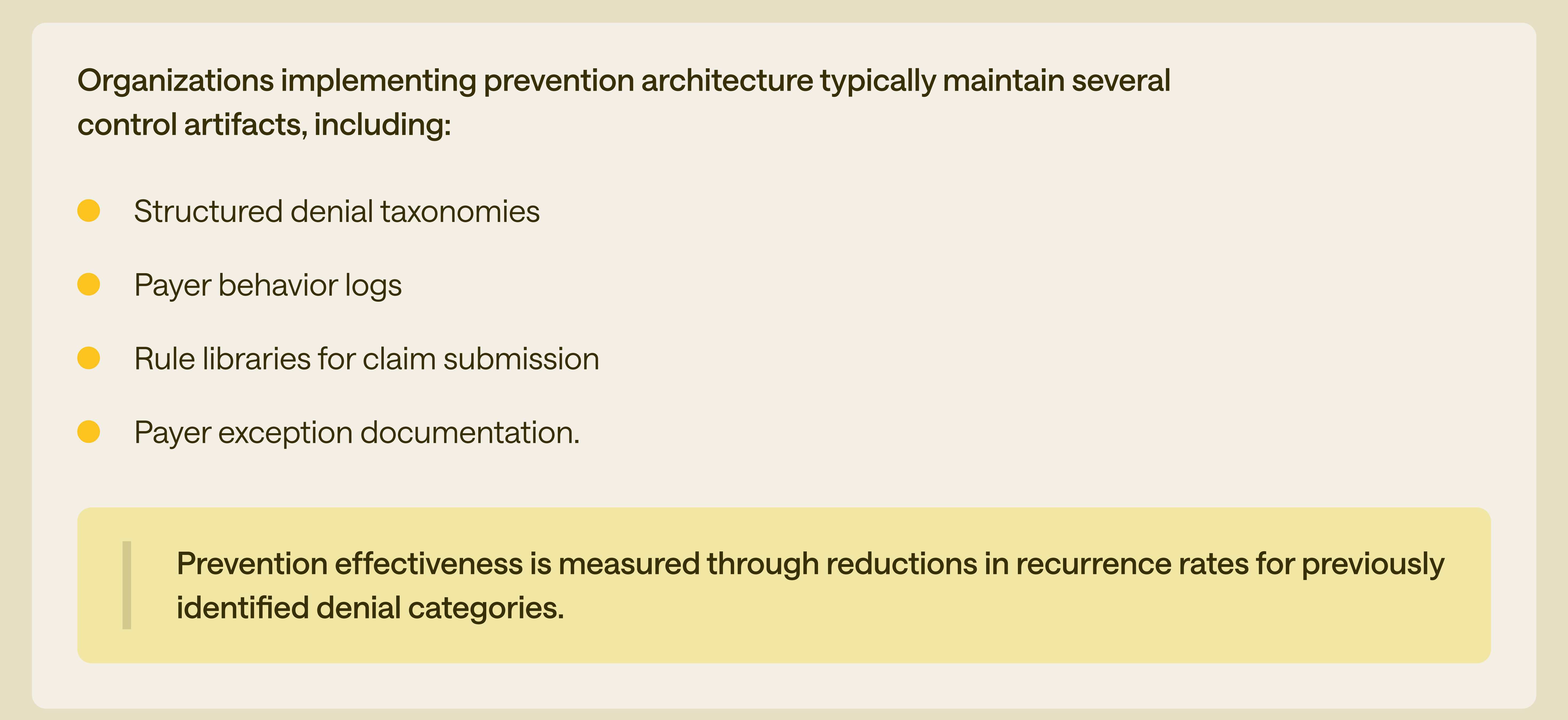
Prevention is often discussed as an aspiration. In denial architecture, it is a **structured control system**.



Prevention operates as a closed-loop process:

- Detection of a denial signal or pattern
- Structured classification of the denial
- Root cause investigation
- Operational intervention
- Monitoring of recurrence rates

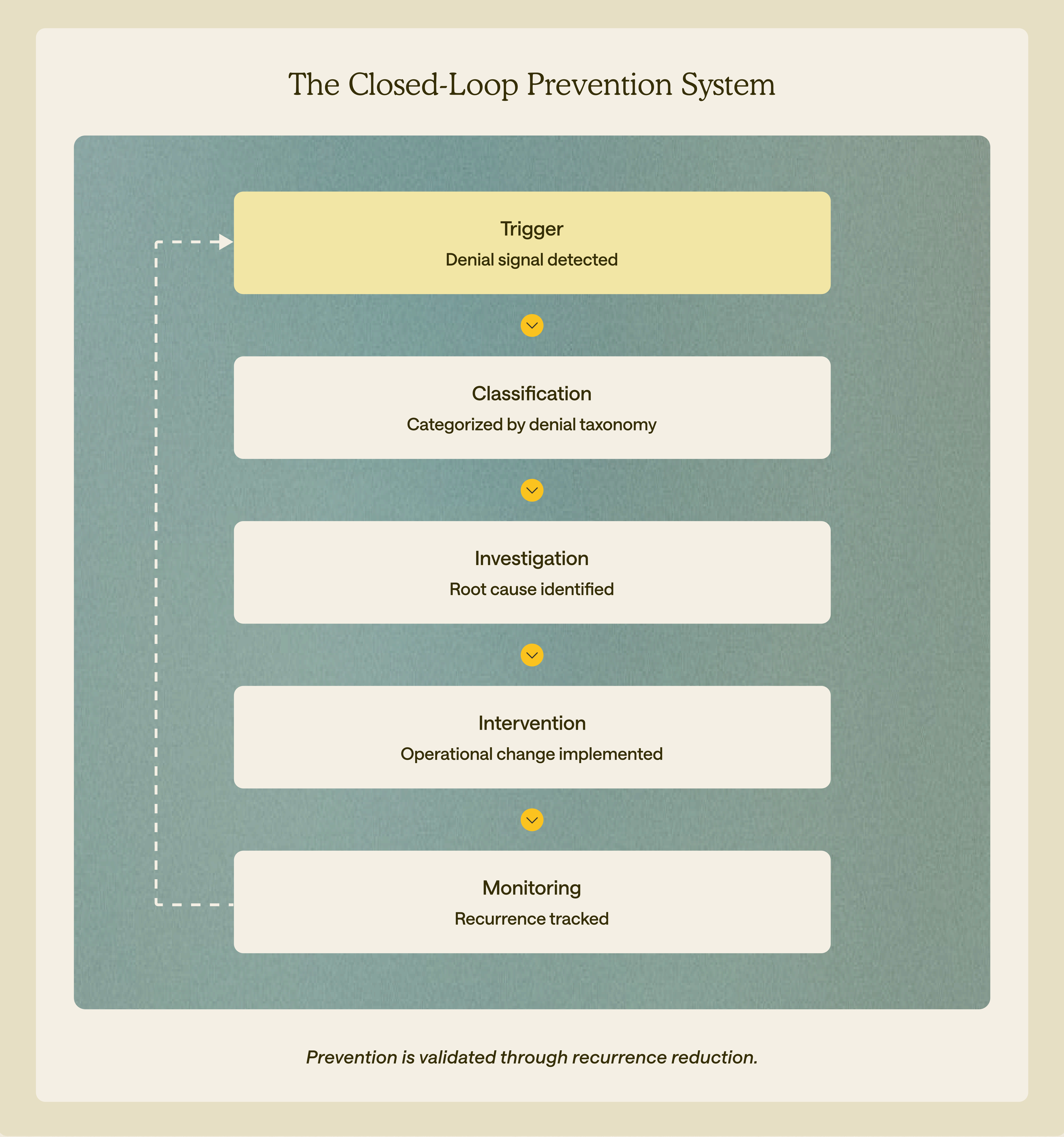
> This feedback loop ensures that prevention efforts are validated rather than assumed.



Organizations implementing prevention architecture typically maintain several control artifacts, including:

- Structured denial taxonomies
- Payer behavior logs
- Rule libraries for claim submission
- Payer exception documentation.

> Prevention effectiveness is measured through reductions in recurrence rates for previously identified denial categories.


The Closed-Loop Prevention System
Trigger
Denial signal detected
Classification
Categorized by denial taxonomy
Investigation
Root cause identified
Intervention
Operational change implemented
Monitoring
Recurrence tracked
Prevention is validated through recurrence reduction.

10

Diagnostic Self-Assessment

Diagnostic Self-Assessment

Healthcare leaders can assess their current denial architecture using a few key questions.

01 What percentage of denials have a documented root cause?

02 Do you track recurrence rates by root cause category?

03 Are denial patterns connected to financial forecasting?

04 Do operational changes trigger structured prevention rules?

05 Can finance explain revenue variance using denial intelligence?

> Organizations answering “yes” to most of these questions typically operate with mature denial architecture.
>
> Those answering “no” are likely operating in reactive recovery mode.

11

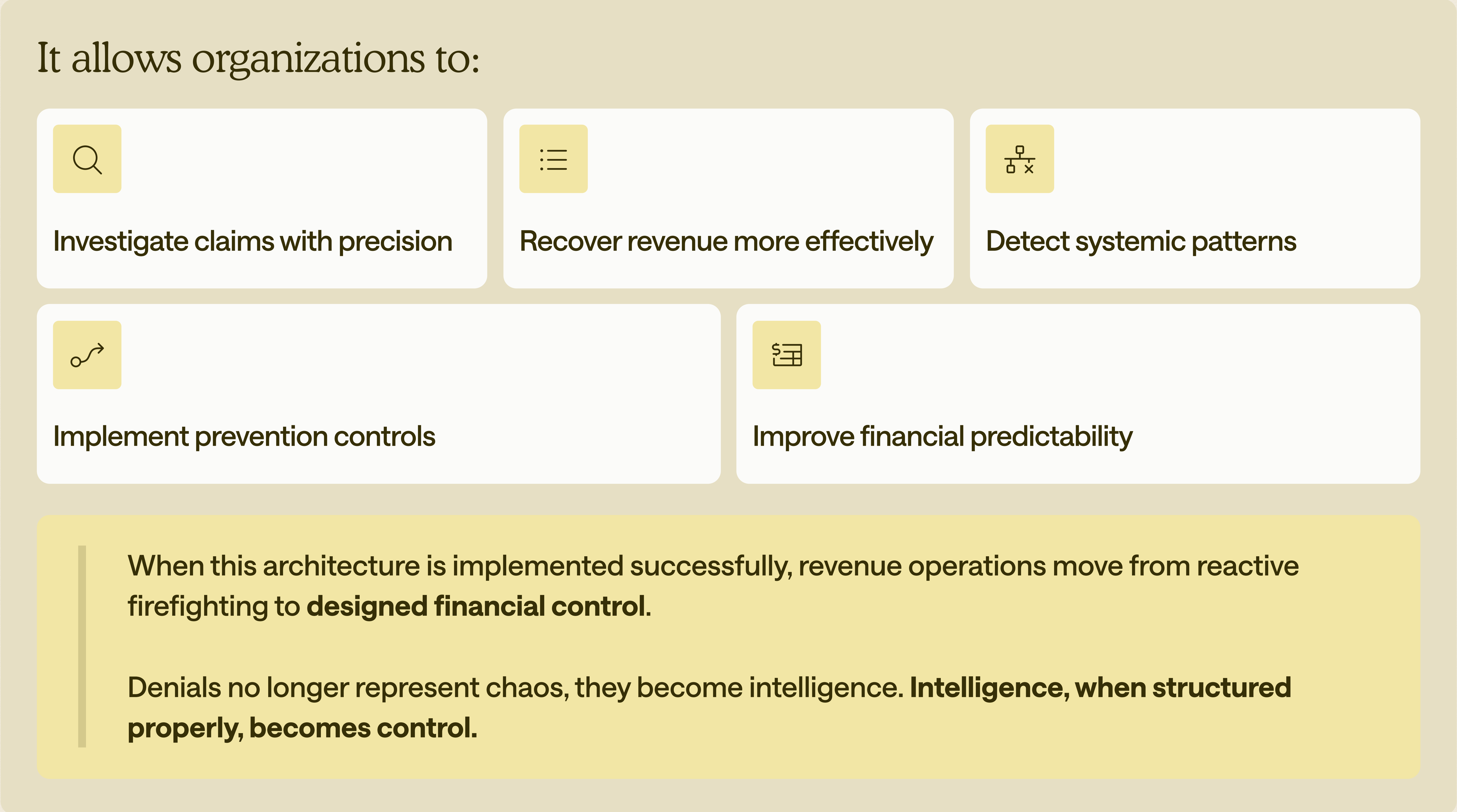
Designing Control into Revenue

Designing Control into Revenue

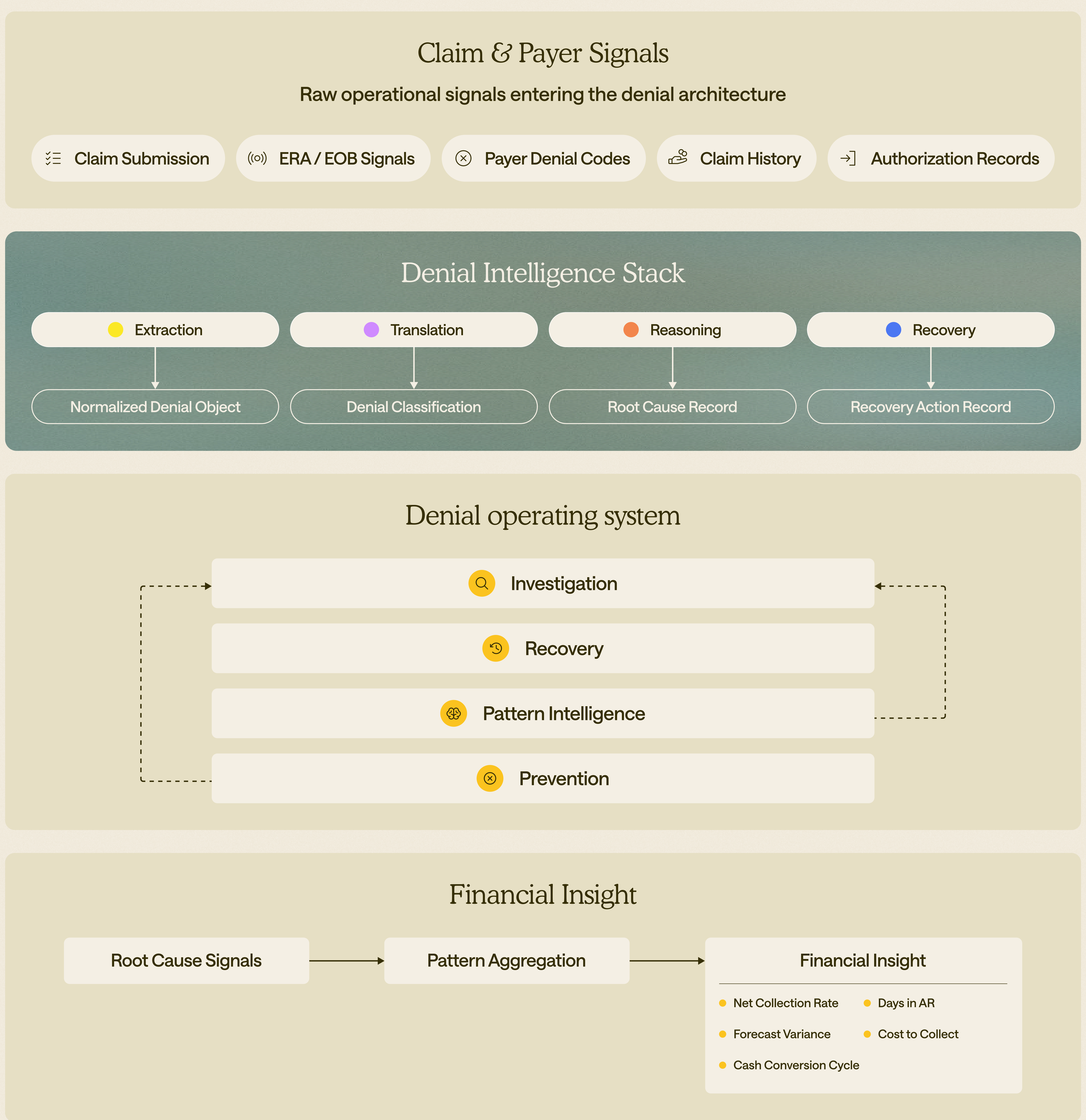
Healthcare revenue systems will continue to grow more complex. Payer rules change frequently. Documentation standards evolve. New payment models introduce additional variability.

Organizations cannot eliminate complexity, but they can design systems that make complexity visible and manageable.

Denial architecture transforms denials from operational frustration into structured financial signals.



The Denial Architecture Model



Denial architecture converts payer signals into operational control and financial predictability.

12

About Joyful Health

Joyful Health provides a Denial Intelligence & Recovery Infrastructure layer for insurance-driven healthcare organizations.

Operating directly within existing revenue systems, Joyful investigates, appeals, follows up on, and resolves denials and unpaid claims end-to-end. The team assesses the entire unpaid claims population - not just the easiest cases - and systematically pursues every viable recovery opportunity. Through structured claim-level investigation and documentation, fragmented payer signals become clear operational insight and financial visibility.

By combining investigation, recovery execution, and structured denial intelligence, Joyful helps revenue cycle and finance leaders recover revenue, reduce recurrence, and improve forecast predictability - without disrupting existing workflows. The result is calmer revenue operations, clearer financial signals, and greater control over the systems that drive payment.