

From legacy estate to production on AWS

How AWS Transform and Thoughtworks AI/works™ combine into one repeatable, MAP-aligned modernization process

/thoughtworks

Design. Engineering. AI.

Table of contents

Executive summary 3

What the combined approach delivers 4

AWS Transform capabilities in the process 4

Evidence flows up, decisions flow down 5

The combined process 8

Aligned to the AWS Migration Acceleration Program 9

Loading AWS Transform output into the graph 10

Assurance through two independent extractions 10

Which platform leads each disposition 11

Scaling Thoughtworks patterns with Custom 12

Testing: equivalence and new behavior 12

Delivery considerations 13

Getting started 14

Sources 15

Author 16



Executive summary

Legacy modernization asks two different kinds of question. Some have right answers: what a COBOL program does, which jobs read a dataset, whether new Java code behaves exactly like the code it replaces. Others are judgments: which capabilities matter, what to retire and what the target product should do.

Thoughtworks combines two platforms so that each handles the questions it's built for. AWS Transform is the engine for analysis and transformation. It does the deterministic, repeatable work at scale: code analysis and business-rule extraction, legacy test capture and functional-equivalence checks, COBOL-to-Java refactoring, Custom transformations, rehost and .NET migration, infrastructure as code and continuous modernization.

Thoughtworks AI/works™ is the decision layer and system of record around it. It keeps a knowledge graph of the whole application estate, scores business value and records a disposition per capability, and maintains the Super Spec: legacy behavior plus new requirements, which drives forward engineering for capabilities worth reimagining or rebuilding. A thin Thoughtworks integration layer joins the two, so AWS Transform output lands in the graph with its provenance intact.

The result is one repeatable, MAP-aligned process that takes a legacy estate to production on AWS, in which every decision is traceable to the customer's own code.

What the combined approach delivers

AWS Transform runs across the lifecycle, and every decision is traceable to the customer's own code.

1. **Faster, evidence-based discovery.** MAP Assess runs on measured AWS Transform analysis rather than interviews, producing a disposition and business case in three weeks.
2. **AWS Transform across the lifecycle.** AWS Transform agents run in seven of the eight stages, from assessment and rule extraction through refactoring, testing and continuous modernization.
3. **AWS-native outcomes.** Beyond refactoring, capabilities worth reimagining are rebuilt as AWS-native services from a traceable specification.
4. **Assurance customers can audit.** Line-level traceability from AWS Transform, cross-checked by a second, independent extraction, gives regulated customers the evidence they need.

AWS Transform capabilities in the process

The joint process relies on eight AWS Transform capabilities. The June 2026 Reimagine release is central: deterministic slicing with line-level traceability and a completeness check.

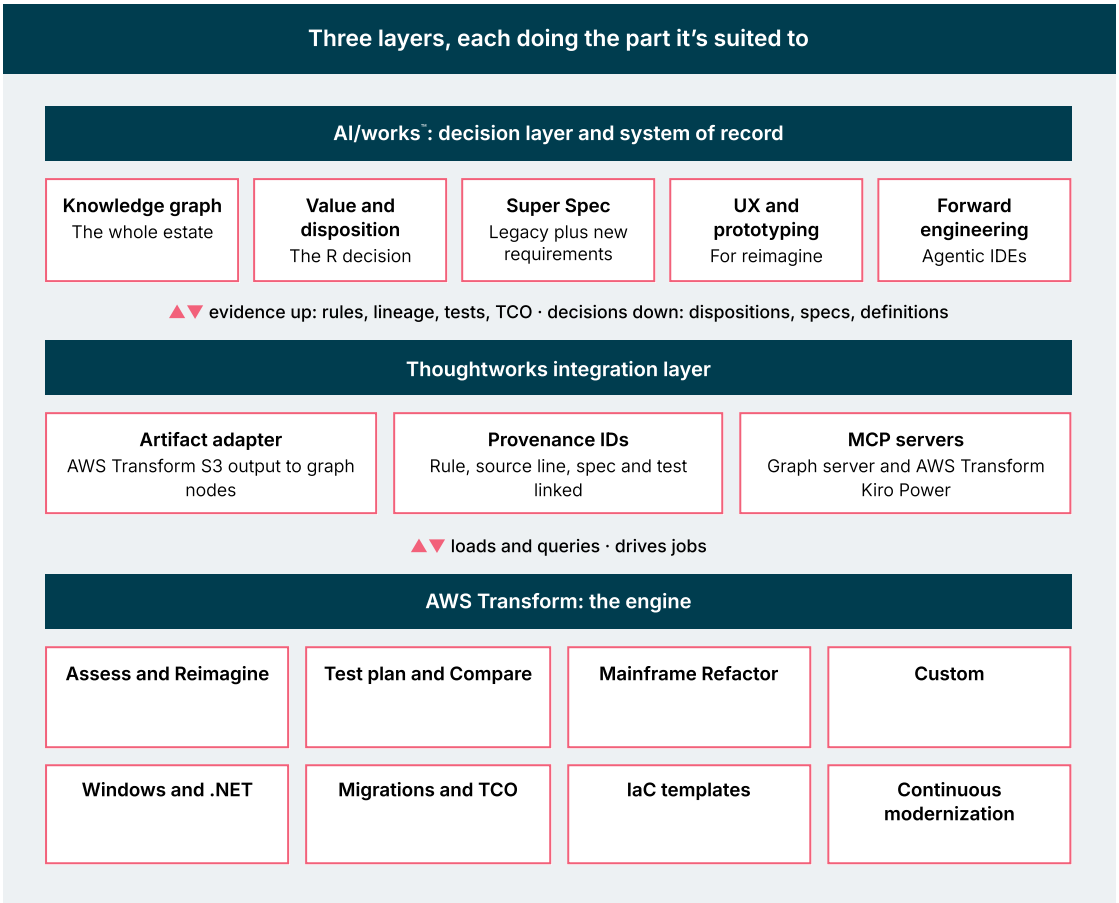
Capability	What it produces	Role in the joint process
Mainframe Assess	Business function catalog; complexity; self-contained vs. shared-datastore functions	Scoping, wave planning and disposition evidence
Mainframe Reimagine	Rules with evidence packs and exact source lines; EARS requirements; lineage; data dictionary; completeness check	Primary input to the knowledge graph and the legacy layer of the Super Spec
Mainframe testing	Test plans; JCL test-data capture (files, VSAM, DB2); automation scripts; bit-level Compare	Functional-equivalence harness
Mainframe Refactor	Deterministic COBOL-to-Java conversion with a test landing zone	Lead tool for the refactor disposition
Custom	Natural-language transformation definitions run across many repositories	Thoughtworks patterns applied at portfolio scale
Windows and .NET	.NET Framework to modern .NET; SQL Server to Aurora PostgreSQL	Lead tool for the replatform disposition

Capability	What it produces	Role in the joint process
Migrations and assessment	Discovery, TCO and what-ifs, EC2 sizing, landing zone, rehost plans	Business case and the rehost disposition
Continuous modernization	Scheduled tech-debt, CVE and readiness scans with fix pull requests	Keeps the new estate current after go-live

Source: AWS Transform documentation and announcements, March to June 2026. See Sources.

Evidence flows up, decisions flow down

The architecture has three layers. AWS Transform sits against the legacy estate and the customer's AWS accounts. AI/works™ sits above it with the knowledge graph and the Super Spec. The Thoughtworks integration layer between them is built once and reused on every engagement.



Evidence flows up from AWS Transform into AI/works™. Decisions flow down from AI/works™ into AWS Transform jobs.

Design principles behind the division of labor

Each platform does the part it's structurally suited to. Nothing is built twice, and every decision can be traced to evidence.

- **Determinism in the engine.** Parsing COBOL, JCL and copybooks, slicing rules and converting code have right answers. AWS Transform provides them with static analysis and line references.
- **Judgment in the decision layer.** Which capabilities matter, what to retire and what the target product should do are business decisions. AI/works™ records them against evidence.

- **Memory in the graph.** Each AWS Transform job produces a snapshot. AI/works™ keeps a model of the estate that persists across jobs, waves and years.
- **Consistent with AWS guidance.** AWS has published a pattern pairing AWS Transform reverse engineering with AI-assisted forward engineering and human review of specifications. The joint process follows that pattern.



The combined process

The value chain runs in eight stages, from a three-day pre-discovery to running code. AWS Transform contributes in seven of them. Through discovery, evidence flows from AWS Transform into AI/works™. At the build, AI/works™ decisions start AWS Transform jobs.

Stage	AI/works™	AWS Transform
00 Pre-discovery 3 days	Agentic cost calculator, cost of doing nothing, first capability sketch	Migration assessment TCO from RVTools or discovery data; readiness and tech-debt scans on Java, .NET and Python repositories
01 Discovery Code to graph	Graph schema and views; extraction for non-mainframe code; capability map	Assess and Reimagine analysis: dependency graph, function catalog, rules with source lines, lineage, data dictionary
02 Research Users and domain	User, market and domain research, tested against real legacy behavior in the graph	Consultant-led stage; no AWS Transform activity
03 Rationalize The R decision	Value scoring, domain boundaries, disposition per capability, wave plan	Coupling map (shared datastores), complexity and dependency counts per function
04 New requirements The Super Spec	New requirements, non-functional needs, UX and target architecture. The Super Spec is the single authority	Legacy requirements in EARS with acceptance criteria, each linked to source lines
05 New TCO Business case	Cost of change per disposition, including the cost of reimagining versus refactoring	Infrastructure TCO, licensing and what-if purchase options
06 MVM build 3 to 9 months	Forward engineering for reimagine and rebuild; AWS Transform artifacts reach coding agents through MCP	Refactor, Custom, rehost and .NET jobs; equivalence tests; IaC templates

Stage	AI/works™	AWS Transform
07 Run After go-live	Graph refresh, scan policy, spec-to-code drift checks	Continuous modernization on the new code: debt and CVE scans with fix pull requests

Aligned to the AWS Migration Acceleration Program

The Thoughtworks 3/3/3 engagement model maps directly onto MAP phases, with AWS Transform running in each.

Pre-discovery · 3 days	**Rough order of magnitude (±50%) that supports MAP Assess approval** AWS Transform: assessment and readiness scans
▲ then	
MAP Assess · 3 weeks	**Disposition per capability, project plan, business case and updated TCO** AWS Transform: Mainframe Assess and Reimagine analysis; infrastructure TCO
▲ then	
MAP Mobilize · 3 to 9 months	**First slice in production: rehost, refactor or reimagine by capability** AWS Transform: Refactor, Custom, .NET, migrations, equivalence testing, IaC
▲ then	
Run · ongoing	**Estate kept current; next waves planned from the graph** AWS Transform: continuous modernization

Repeatable across the estate: each wave reuses the graph, the integration layer and the Custom definitions built in earlier waves.

Loading AWS Transform output into the knowledge graph

AWS Transform writes a folder per capability to Amazon S3. The Thoughtworks artifact adapter turns that output into graph nodes and edges, keeping AWS Transform rule IDs and source references intact so every downstream artifact stays traceable.

Node	How it connects
Rule *(from AWS Transform)*	Evidenced by a source line, verified by a legacy test, reads and writes data entities, and specified as a Super Spec requirement
Capability *(the join)*	Contains rules, is assigned a disposition (R decision plus value) and is realized by a target service
Spec requirement *(AI/works™)*	Traced to the rule it specifies and extended by new requirements
Target service *(AI/works™)*	Realizes a capability, so any new service traces back to the legacy code it replaces

Integration principles

- **AWS Transform identifiers preserved.** Rule IDs and source references are stored as they are, never re-keyed.
- **Many-to-many mapping.** AWS Transform business functions map to domain capabilities explicitly.
- **Versioned loads.** Each AWS Transform run is a snapshot, so re-runs show what changed.
- **Open access through MCP.** Coding agents query the graph and AWS Transform artifacts in one session.

Assurance through two independent extractions

On mainframe code, a deterministic extraction and an agentic extraction are compared. Agreement is accepted, and differences go to review. Agreement between two independent methods gives stronger assurance than either alone, and it concentrates expert review on the small set of differences.

AWS Transform Deterministic slicing	+	AI/works™ Agentic extraction	→	Reconcile Match on source lines

Result	What happens
Both agree	Accepted into the Super Spec automatically
AWS Transform only	Used as a coverage check of the agentic extractor
AI/works™ only	Confirmed against source; often inferred or unreachable code
Conflict	A Spec Steward reviews both evidence packs

Every MAP Assess readout reports the agreement rate, and the list of reviewed differences with their resolution.

Which platform leads each disposition

AWS Transform leads five dispositions directly and supports the rest with evidence and testing. AI/works™ leads where business judgment or new design is required.

Disposition	AI/works™	AWS Transform	How it runs	Typical MVM
Retain	**Lead**	Assist	Decision recorded in the graph; continuous modernization keeps it current	n/a
Retire	**Lead**	Assist	Graph shows no live callers; AWS Transform lineage confirms nothing reads the data	Weeks
Rehost	Assist	**Lead**	Migrations agent (VMware, Hyper-V, Nutanix to EC2 or containers)	~3 months
Replatform	Assist	**Lead**	Windows agent for .NET and SQL Server; mainframe replatform through AWS partners	~3 months
Refactor	Assist	**Lead**	Mainframe Refactor, COBOL to Java; Compare proves equivalence	~6 months
Rearchitect	Assist	**Lead**	Custom running Thoughtworks-authored definitions across repositories	~6 months
Rebuild	**Lead**	Assist	Generated from the Super Spec; AWS Transform tests prove legacy equivalence	6 to 9 months
Reimagine	**Lead**	Assist	AWS-native services designed from the Super Spec; AWS Transform artifacts through MCP	~9 months
Replace (COTS)	**Lead**	Partner	Gap analysis of legacy capabilities against the package	Varies

Disposition	AI/works™	AWS Transform	How it runs	Typical MVM
Augment	Lead	Assist	New AWS-native services around a retained core	~3 months

Scaling Thoughtworks patterns with AWS Transform Custom

Recurring modernization patterns are written once as Custom definitions and applied across the customer's portfolio. Each engagement improves the definition.



Initial candidate patterns:

- Strangler-fig seams and anti-corruption layers
- Event extraction from batch jobs
- Standard Spring and .NET upgrade paths
- Logging, configuration and security baselines for AWS

Testing: equivalence and new behavior

Functional equivalence is the top priority for rewrites, and new behavior needs its own tests. Each platform owns one side.

Functional equivalence

AWS Transform · must behave exactly like the legacy system

1. Test plan from program, job and scheduler dependencies
2. JCL scripts capture real data from files, VSAM and DB2
3. Automation scripts set up, run and compare
4. Compare checks output bit by bit: files, rows, PDFs

New behavior

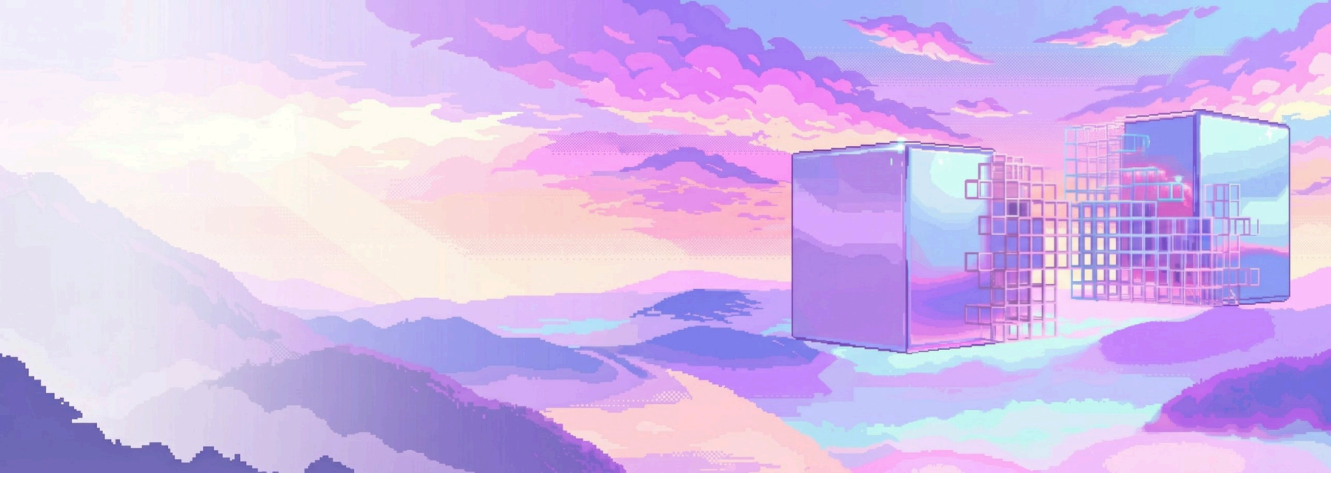
AI/works™ · new requirements and reimagined flows

1. Acceptance criteria in the Super Spec
2. Tests generated from each criterion
3. Each test linked to its requirement node
4. Coverage reported from the graph

Retained rules in reimagined services get both: an AWS Transform equivalence test for retained behavior and a spec test for the new surface.

Delivery considerations

1. **Scope of mainframe Reimagine:** it covers z/OS COBOL and PL/I. For Java, .NET and other stacks, AI/works™ leads capability extraction and AWS Transform leads transformation.
2. **One source of truth:** the Super Spec is the authority, and AWS Transform output is linked evidence, so the two never diverge.
3. **Security and data approvals:** AWS Transform runs in the customer's AWS Region. Connector approvals start in pre-discovery so MAP Assess isn't delayed.
4. **Artifact format stability:** the integration layer is versioned and tested against each AWS Transform release.
5. **Roadmap alignment:** regular syncs with the AWS Transform team keep the integration aligned as new capabilities ship.



Getting started

AWS Transform analyzes and transforms. AI/works™ decides and remembers. Together they take a legacy estate to production on AWS, with every decision traceable to the code it came from and every wave building on the last.

Customers can start small and fund each step through MAP:

1. **Pre-discovery (3 days):** a rough order of magnitude, built on AWS Transform scans, that supports MAP Assess approval.
2. **MAP Assess (3 weeks):** a disposition per capability, a project plan, a business case and an updated TCO, with the extraction agreement rate reported.
3. **MAP Mobilize (3 to 9 months):** the first slice in production, by rehost, refactor or reimagine.

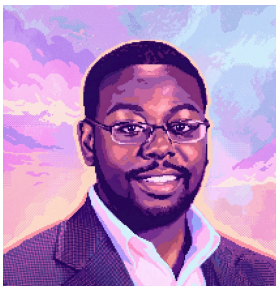
AWS Transform analyzes and transforms. AI/works™ decides and remembers.

To plan a pre-discovery for your estate, contact us today.

Sources

1. AWS Transform for mainframe: traceable reimagine workflow, AWS What's New, June 2026
2. Reimagine mainframe applications with traceability and speed using AWS Transform, AWS Migration and Modernization Blog
3. Reimagining mainframe applications with AWS Transform and Claude Code, AWS Migration and Modernization Blog
4. Accelerating mainframe modernization testing with AWS Transform, AWS Migration and Modernization Blog
5. AWS Blu Insights is now AWS Transform for mainframe refactor, AWS What's New, March 2026
6. AWS Transform FAQs
7. AWS Transform Custom documentation

Author



Rickey Zachary

Global Lead of Platform Engineering, Thoughtworks

We are a global technology consultancy that delivers extraordinary impact by blending design, engineering and AI expertise.

For over 30 years, our culture of innovation and technological excellence has helped clients strengthen their enterprise systems, scale with agility and create seamless digital experiences.

We're dedicated to solving our clients' most critical challenges, combining AI and human ingenuity to turn their ambitious ideas into reality.

thoughtworks.com

/thoughtworks

Design. Engineering. AI.

Last updated December 2026
© Thoughtworks, Inc. All Rights Reserved.