

OTC Operating Model

A governance architecture for reliable execution in high-volume work environments

More Work Arrives Than Can Ever Be Done

Requests, decisions, information, interruptions, possibilities. None of it waits for what is already in progress.

Volume like this is not an exception anymore. It is the continuous operating condition of most professional environments.

Against this, execution capacity is finite. There are only so many hours in which work can actually be done, and no amount of effort changes that arithmetic.

An unbounded stream of demands is meeting a bounded capacity to execute.

The Problem Is Not the Workload

The usual response to unbounded work is the pursuit of better prioritization, more discipline, more tools, another method. These approaches optimize behavior inside an environment that has no defined structure, and so nothing fundamental changes.

Commitments are made without knowing the limits of the work environment in which they must be executed. New work enters and leaves without structure. Urgency repeatedly displaces what had already been intended for execution.

The underlying failure is structural. Incoming work is not governed. Decisions are deferred instead of made. Execution, possibilities and material are allowed to coexist without reliable boundaries.

What is missing is not effort or discipline. It is an operating structure that governs execution.

What Reliable Execution Requires

If the failure is structural, the requirements can be stated structurally. For execution to remain reliable while demands continue to arrive, four conditions must hold:

A bounded Execution Environment

There must be a defined environment within which execution is governed, and a clear boundary between what is inside that environment and what remains outside it.

Governed entry

Entry into the Execution Environment must occur through defined routes.

Continuous governance

Everything that enters remains under governance for as long as it remains inside.

Defined exit

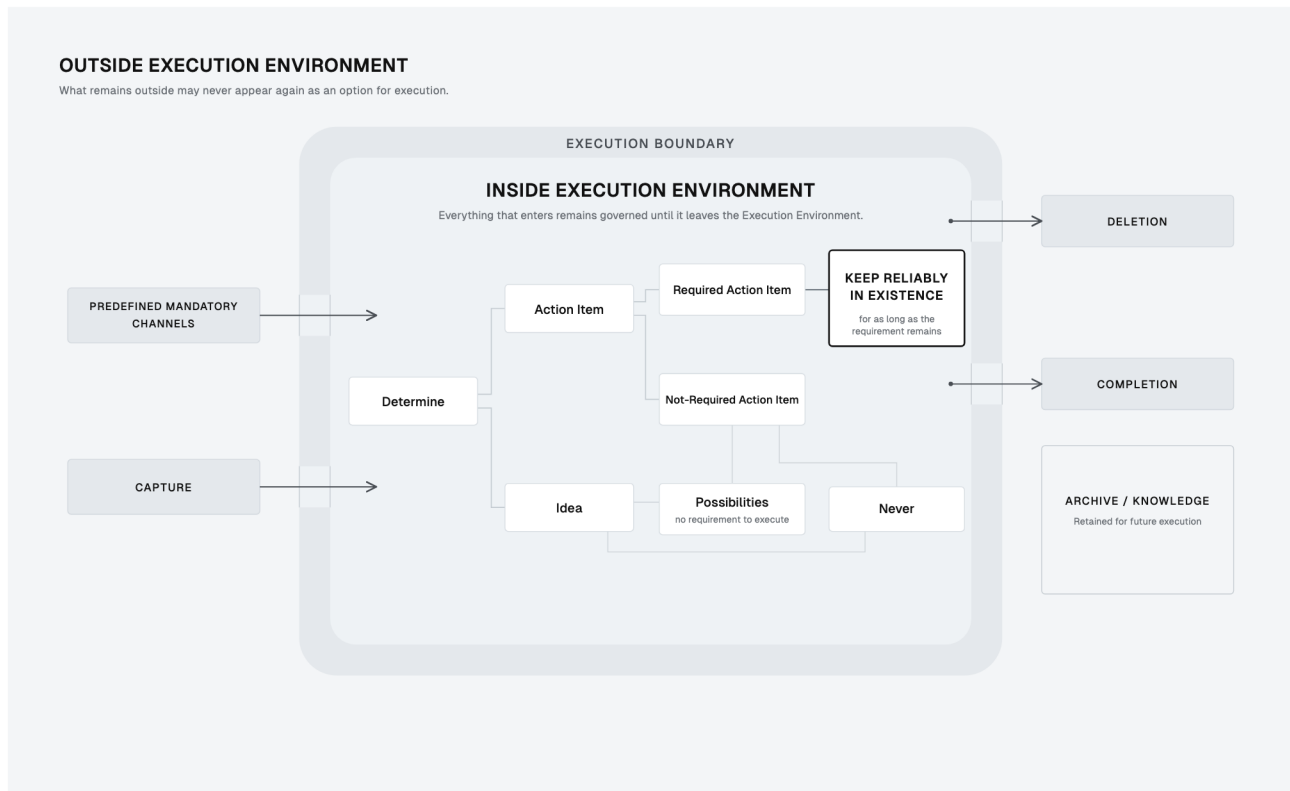
Something leaves only through completion or deletion.

None of these conditions depends on a particular tool, method or personality. They are structural requirements of governable execution. The question is whether an architecture establishes them and continues to hold them as volume increases.

One Governed Execution Environment

The OTC Operating Model establishes these conditions as one architecture.

At its center is the Execution Environment: a governed environment, one per person, in which everything that person will execute is held. Around it runs the Execution Boundary, the defined line between what has entered and what remains outside. Work intended for execution does not live in side lists, parallel systems or someone's memory. It lives in one governed place, or it is not going to happen.



What enters is determined

The boundary governs entry. Something enters the Execution Environment either because it arrives through a predefined mandatory incoming communication channel or because it is explicitly captured into it. Determine distinguishes Ideas from Action Items and establishes what requires execution.

Action Items are governed

Every Action Item inside the Execution Environment is governed. Required Action Items are held in execution structures that keep them reliably in existence. Not-Required Action Items remain available without becoming commitments.

Execution is bounded

The Operating Model does not attempt to make everything executable. It establishes what is available for execution and exposes the limits within which execution decisions are made.

Tools operationalize the architecture

Capture tools, calendars, lists and archives support the architecture. They are mechanisms through which the Operating Model is operated; they are not the Operating Model itself.

The architecture does not depend on particular software. Tools operationalize the Operating Model; they do not define it.

Four Governing Principles

The architecture is operated through four governing principles.

Face the Beast

Maintain contact with the full reality of what requires attention.

Keep Actions Reliably in Existence

Once an Action Item exists, hold it in a structure that keeps it available until it is completed or deliberately removed.

Do Complete Work

Whenever work is taken on, carry it to a defined complete condition rather than leaving it in an undefined state.

Work with Integrity

Execute in accordance with the commitments and structures that have been established.

Why It Holds as Volume Increases

The Operating Model does not reduce the amount of work that arrives. It changes the conditions under which that work is governed.

As volume increases, the architecture does not require additional complexity. The same boundaries, entry rules, governance and completion conditions continue to apply.

The higher the work volume, the more strictly the Operating Model is followed.

Execution Is Individual. Governance Extends Across People

Execution always occurs at the level of the individual. Every person operates within their own Execution Environment and governs the work for which they are responsible.

But work does not remain individual. Commitments cross between people, teams and functions. The reliability of one person's execution therefore affects the reliability of others.

The same Operating Model can extend across these boundaries without creating a second architecture. What changes is the context in which governance is established.

This creates three engagement pathways: individual implementation, team installation and organizational installation.

Individual

The Operating Model is implemented in each participant's own work.

Team

The Operating Model is installed across a team, establishing a shared execution architecture while each person continues to operate their own Execution Environment.

Organization

The Operating Model is installed across organizational boundaries where execution reliability depends on commitments moving between people, teams and functions.