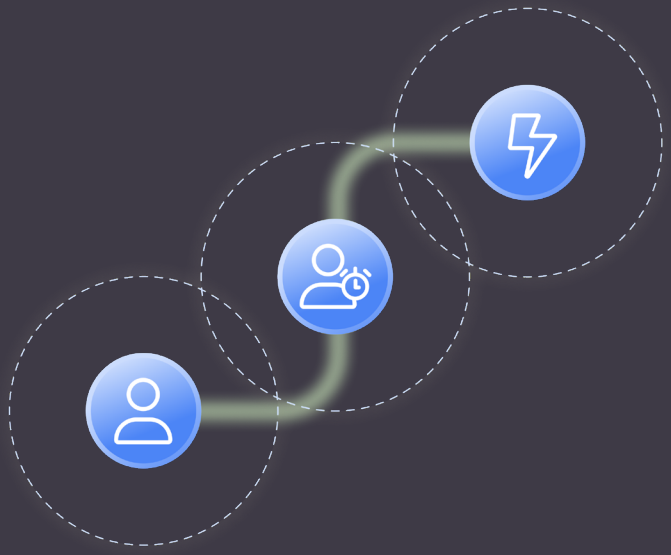
ConductorOne

Access Controls Maturity Model



A step-by-step guide to modernizing identity access controls with ConductorOne

Access control isn't a checkbox—it's a journey. Most teams start out reactive, manually fulfilling access needs, and racing to meet audit requirements. But with the right tools, you can evolve your access management program one step at a time.

Use this guide to identify where you are today, understand what it takes to level up, and see how ConductorOne supports you at every stage.

Level 1: Basic automation

Where most companies start: spreadsheets, watercooler approvals, but some basic automation

What it is:

Most organizations don't start fully manual—but it still feels manual. You might have an IdP, a ticketing system, and Slack or email driving approvals. But provisioning and deprovisioning is still mostly done by hand, access is reviewed only when forced by audits, and over-permissioned users are everywhere.

What it looks like:

→ Access requests via Slack, Teams, Jira, ServiceNow, email, or hallway drive-bys
→ Manual provisioning and deprovisioning from a single IT manager or app owners
→ Excel-based access reviews with incomplete evidence trails
→ Helpdesk tickets dominate routine access requests
→ Limited or no central view of user entitlements
→ Birthright access (like email, Slack) handled automatically, but sensitive apps remain a ticket-driven mess

Why move forward:

→ This is no way to live!
→ Over-permissioned users increase risk
→ Audit prep is painful and reactive
→ IT is bogged down with repetitive access tickets
→ Role bloat is already happening
→ You can't scale access operations without adding headcount

What success looks like:

→ Self-service access for common, low-risk apps
→ Basic approval workflows for sensitive access
→ Real-time provisioning through your IdP or SCIM connections
→ Beginnings of an entitlement inventory
→ Context-driven, documented approvals

What you need:

→ Unified identity and entitlement visibility
→ Self-service access request portal
→ Basic approval logic tied to roles, teams, or app owners
→ Lightweight emergency (break-glass) access workflows

ConductorOne

How ConductorOne helps:

→ Unified Identity Graph to visualize every account and entitlement, even for legacy and on-prem apps
→ Self-service access requests that reduce IT ticket volume by up to 95%
→ Flexible approval workflows that route based on user attributes, managers, app owners, or on-call schedules
→ Emergency access flows to unblock users securely without manual chaos
→ One-click, audit-ready access reviews and evidence capture

How to level up:

Move from “request and forget” to dynamic access: access that’s granted based on user attributes (team, department, title) and automatically expires after a set time. Start reducing standing access by setting time limits on elevated permissions.

Level 2: Role- and Attribute-Based Access

Smarter, faster access: granted when it’s needed based on role or attribute, revoked when it’s not

What it is:

RBAC and ABAC are now combined. Access is granted on the spectrum of birthright, role based, or attribute based to ensure right sized permissions. Just-in-time (JIT) augments access controls to ensure that the most sensitive of permissions are only requested for short periods of time.

What it looks like:

→ Developers request short-lived access to AWS, production, infrastructure, or sensitive systems

→ Approval flows are dynamic, adjusting based on title, team, system, risk level, or other conditions

→ Access is temporary, policy-enforced, and auto-revoked

→ Roles and attributes drive automated birthright or joiner-mover-leaver provisioning of access

→ Account and entitlement provisioning from IdP and some direct connectors

Why move forward:

→ Birthright access likely still bloated

→ Standing access is no longer acceptable

→ You want a central, auditable, “brain” for management entitlements and access control

→ Real-time enforcement and risk driven decision making is critical

What success looks like:

→ All privileged access is JIT

→ Access is granted and revoked automatically

→ Conflicts and SoD violations are flagged proactively

What you need:

→ Dynamic policy engine

→ JIT coverage for critical systems

→ Smart routing and risk-scoring

ConductorOne

How ConductorOne helps:

→ Full support for JIT access, with auto-expiry windows ranging from hours to weeks

→ **AI-generated context with Copilot** to help reviewers make confident, accurate decisions

→ Requestable job function sets let teams request a known group of entitlements as a bundle, without having to remember each individually

→ Automatic revocation ensures no lingering access remains after the job is done

Even traditionally over-permissioned functions can be broken down into temporary bundles that users request only for the time needed—no more guesswork, tickets, or standing access.

Level 3: Context-driven Entitlement Management

Real-time, attribute and policy-driven access controls supplemented with just-in-time access and self service requests. You have a fully automated entitlement engine and provisioning engine.

What it is:

The peak of access control maturity. Access is entirely real-time, immediate, context driven, and risk limited. Identities receive permissions when they need them, for as long as they need them, and no longer. Everything—request, review, provisioning, and revocation—is automated and coordinated.

What it looks like:

→ Zero standing access for sensitive permissions

→ Every access decision is evaluated in real time based on identity, risk, and business logic

→ Requests, approvals, and revocations are handled through fully automated workflows

→ Roles and functions have right sized entitlement bundles for essential access

→ Self-service still exists, but it's used to manage drift—not to drive core birthright and role based access

→ Real-time connectors supporting provisioning for all SaaS, infrastructure, on-prem, and cloud applications

Why move here:

→ Reduce risk of excessive privileges

→ Enable compliance automation, auditability, and access transparency

→ Give developers the flexibility to work securely at speed

→ Scale your access management program with access control managed "as code"

What success looks like:

→ All access is policy-driven and justified

→ Lifecycle automation covers every access flow

→ Drift is corrected automatically via orchestration

What you need:

→ Continuous policy evaluation

→ Integration with HRIS, IdPs, and infrastructure for deep attribute context

→ Developer-ready extensibility via APIs, SDKs, and Terraform

ConductorOne

How ConductorOne helps:

→ **ConductorOne** is the entitlement management and provisioning engine for your organization

→ Access orchestration lets you manage policies, reviews, provisioning, and revocation from a single, unified control layer

→ Manage access “as code” with Terraform. Define all access controls, policies, profile, and entitlement configurations using GitOps.

→ Our unbounded API lets your internal tools interact directly with **ConductorOne**, enabling limitless automation

→ The deepest and most robust out of the box connectors for real-time entitlement and account provisioning and management

→ Ability to support on-prem, hosted, and legacy databases and infrastructure

Even as systems become more automated, self-service continues to serve an important role. While phase 1 and 2 access is help-desk driven, phase 3 and 4 transitions to a world where help desk intake, access review, and provisioning are fully automated and orchestrated via real-time policies.

Wherever you are, just keep moving

Every step forward on the maturity curve strengthens your security, improves compliance, and reduces operational burden. Whether you’re managing tickets in Excel or provisioning access with Terraform, what matters most is forward motion—and **ConductorOne** is built to meet you where you are and take you further.