DevOps and the Future of Change Management

Mike Long @meekrosoft

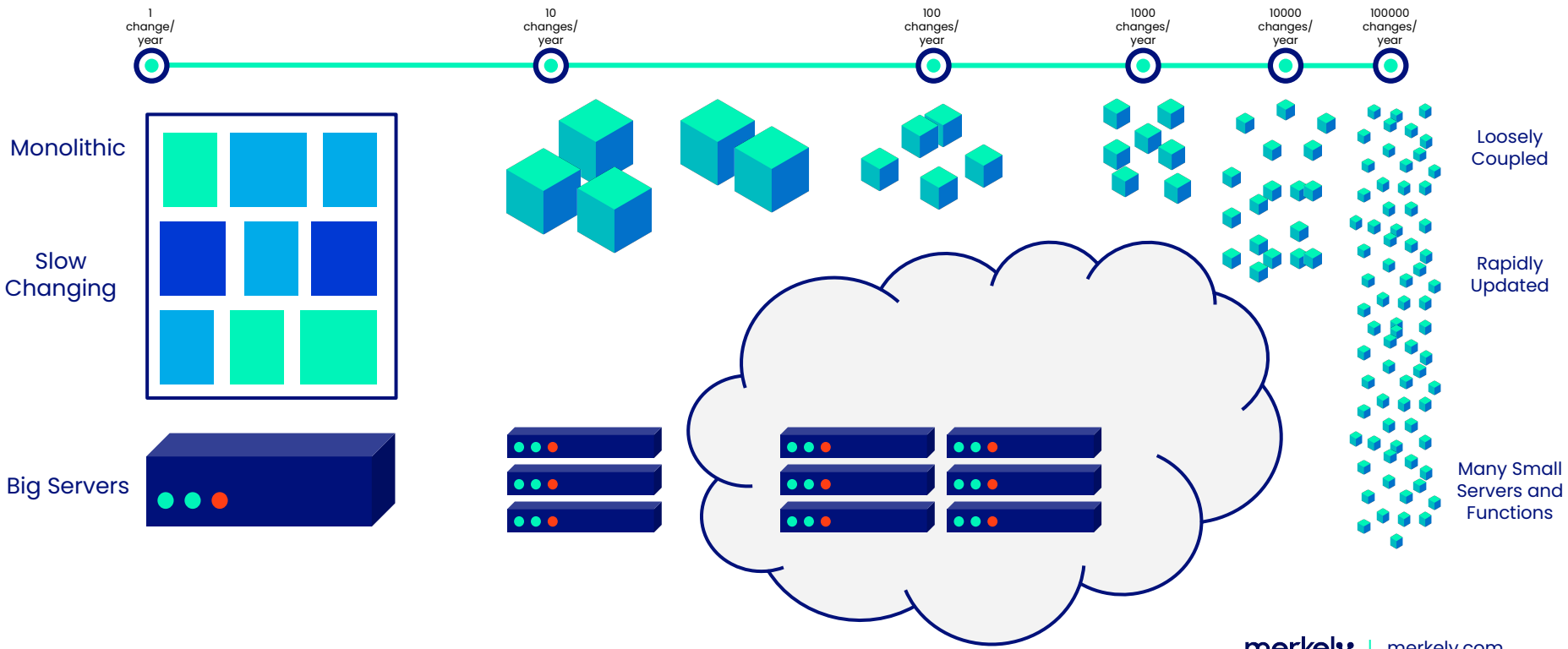
www.merkely.com

merkely

“Ultimately the winners in banking will have the capabilities of a world-class software company”

Rich Fairbank Capital One Founder & CEO

Software is Changing

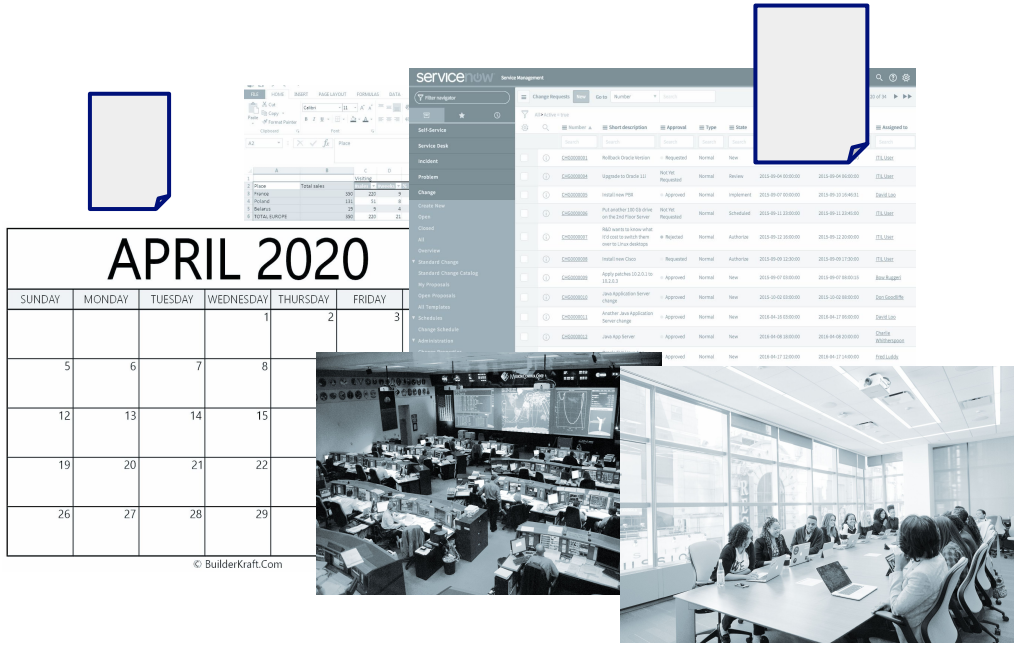


Change Management Must Change

The old ways of managing change **can't keep up** with modern devops teams:

- Change Windows
- CAB Meetings
- Manual Documentation
- Batched Changes

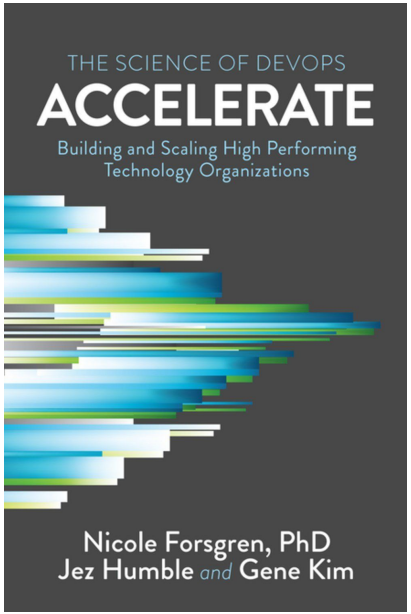
These methods are proven to be ineffective at mitigating risk*



* Forsgren PhD, Nicole. Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations . IT Revolution Press. Kindle Edition.

Change Management Board

We found that **external approvals were negatively correlated with lead time, deployment frequency, and restore time**, and had no correlation with change fail rate. In short, approval by an external body (such as a manager or CAB) simply doesn't work to increase the stability of production systems, measured by the time to restore service and change fail rate. However, it certainly slows things down. **It is, in fact, worse than having no change approval process at all.**



Forsgren PhD, Nicole. Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations . IT Revolution Press. Kindle Edition.

Why is it like this?

APRA

Australian Prudential Regulation Authority

Security in change management

47. APRA envisages that a regulated entity would implement controls to manage changes to information assets, including changes to hardware, software, data, and configuration

(both where the change is planned and in response to an emergency) with the aim of maintaining information security. This would typically include:

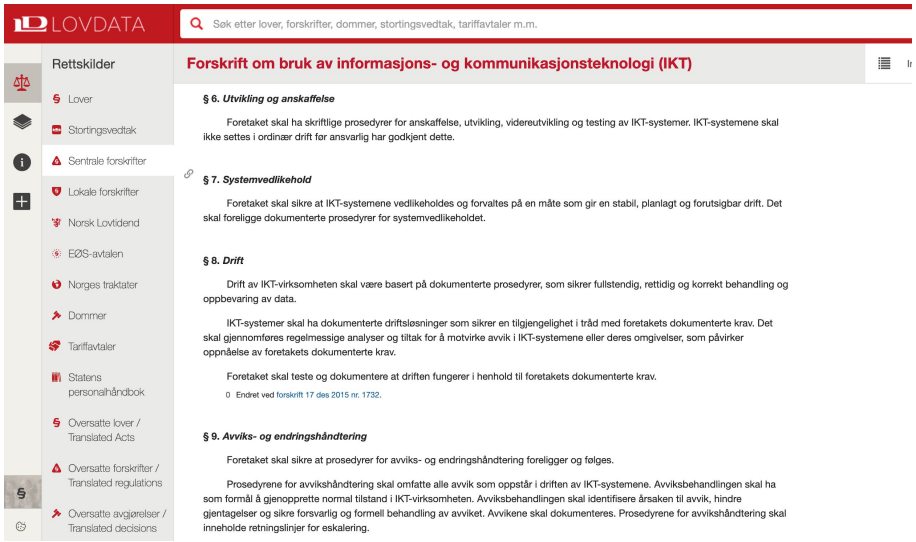
a) security testing (including reviews) to identify vulnerabilities and confirm information security requirements have been met. The nature of testing would be commensurate with the scope of the change and the sensitivity and criticality of the impacted information asset (refer to Attachment H for examples of common testing techniques);

b) approval of changes prior to deployment into the production environment;

c) segregation of duty controls which prevent personnel from deploying their own software changes to production;

d) changes are developed and verified in another environment, sufficiently segregated from production so as to avoid any compromise of information security;

What does an audit look like?



Regulated Industries are obliged to manage change.

- What is your software development process?
- How do you ensure conformance to the process? Can you prove it?
- What software is currently in production?
- What changes were in the last release?
- What was deployed on 13/02/2020?
- Who accepted and signed off on the risk?

What is Change Management Anyway?

Governance Framework

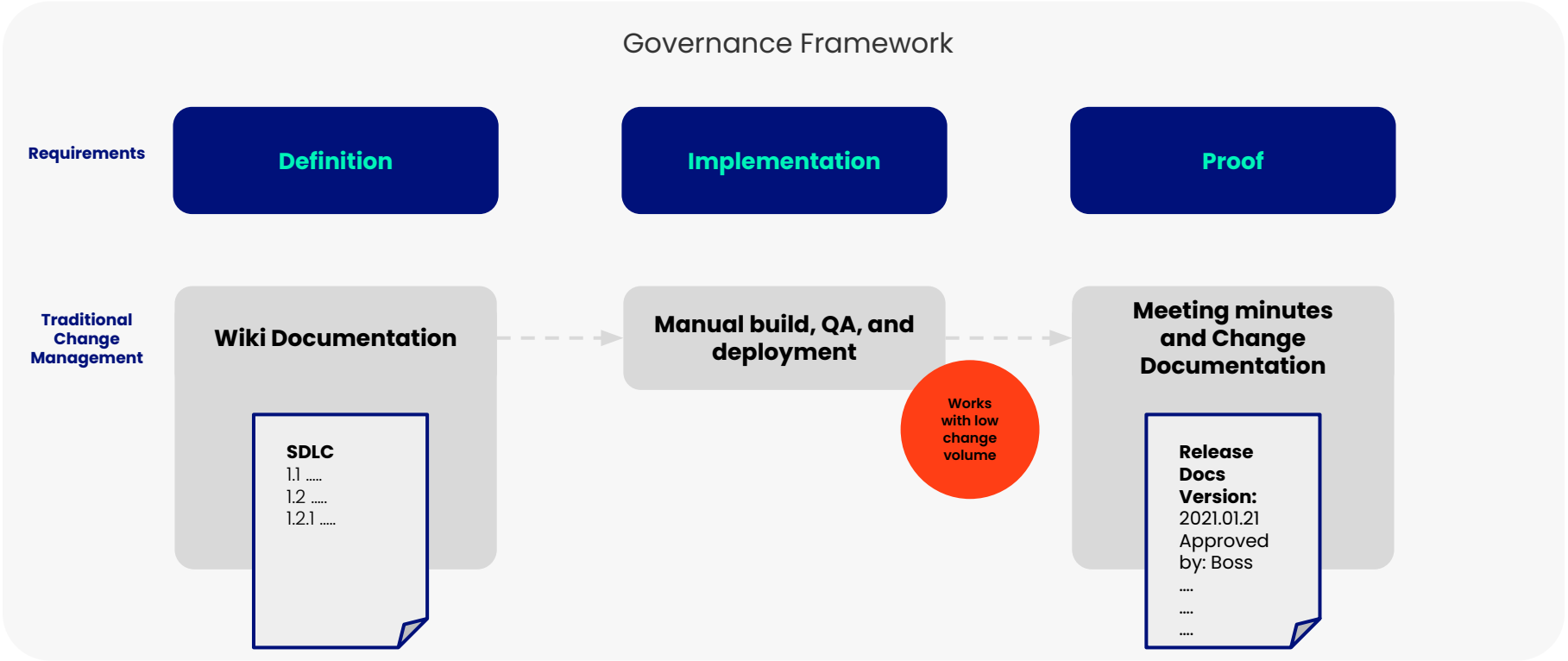
Requirements

Definition

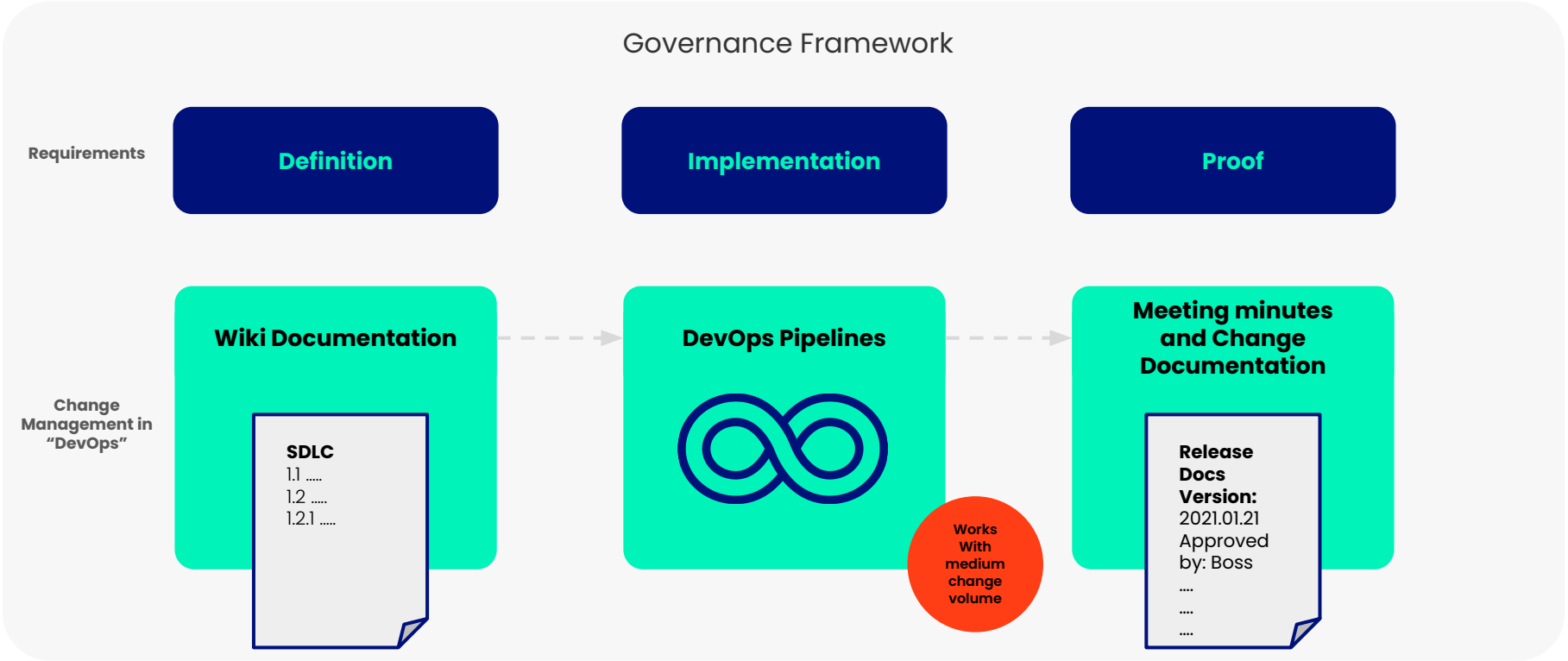
Implementation

Proof

Traditional Change Management



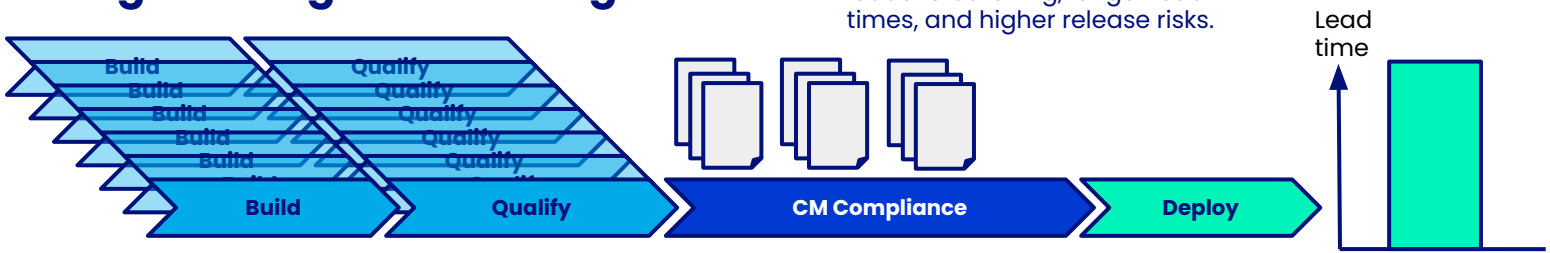
Change Management in “DevOps”



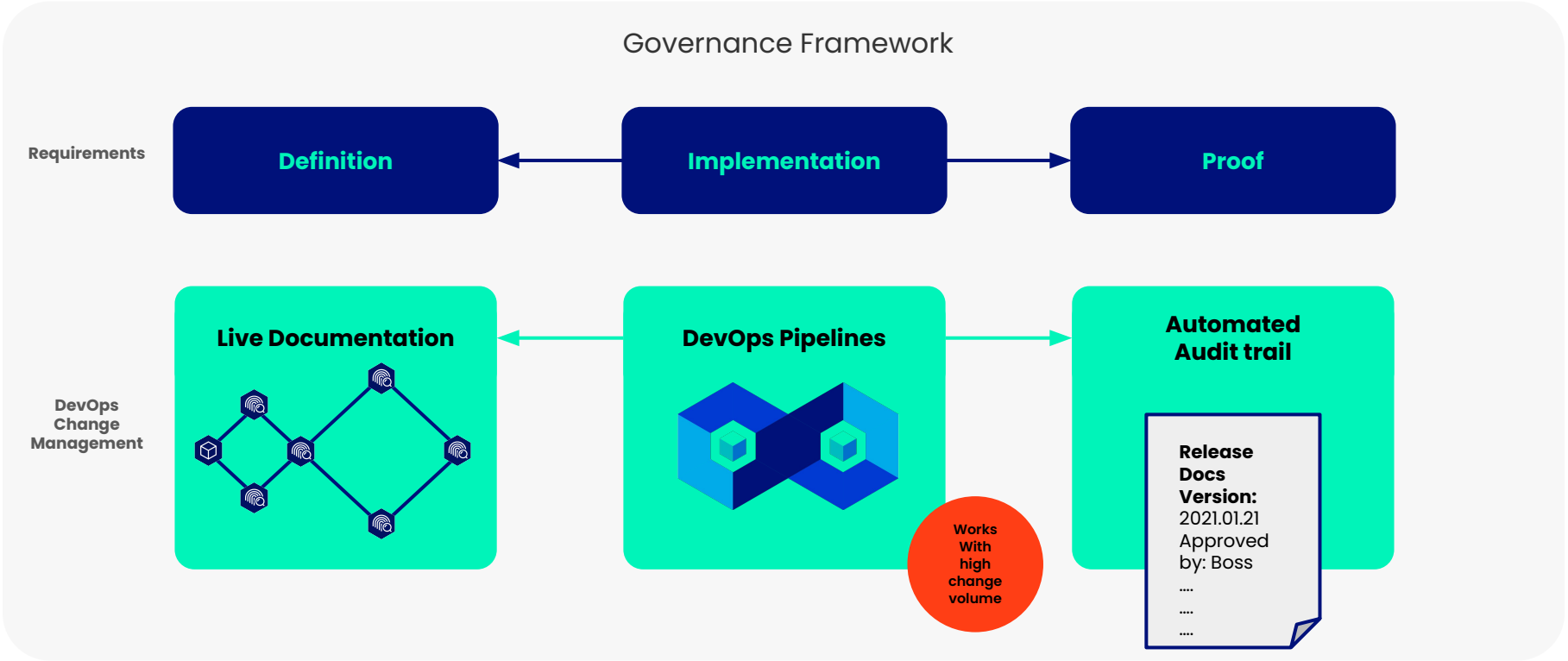
Lead Time for Changes with “DevOps”

Change Management as a gate

High ceremony CM compliance leads to batching, longer lead times, and higher release risks.

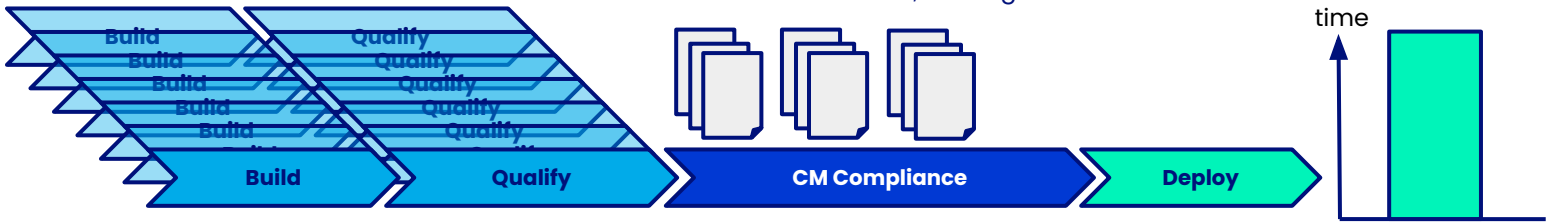


DevOps Change Management



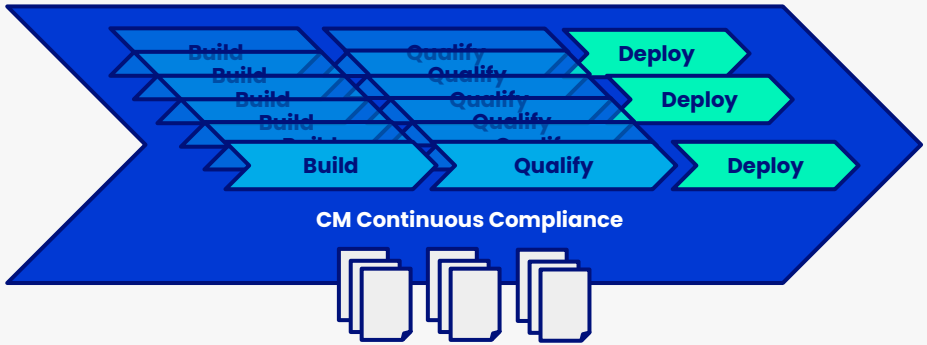
Lead Time for Changes with *DevOps*

Change Management as a gate

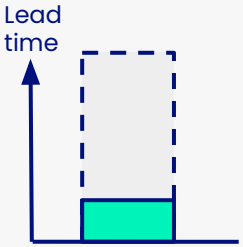


High ceremony CM compliance leads to batching, longer lead times, and higher release risks.

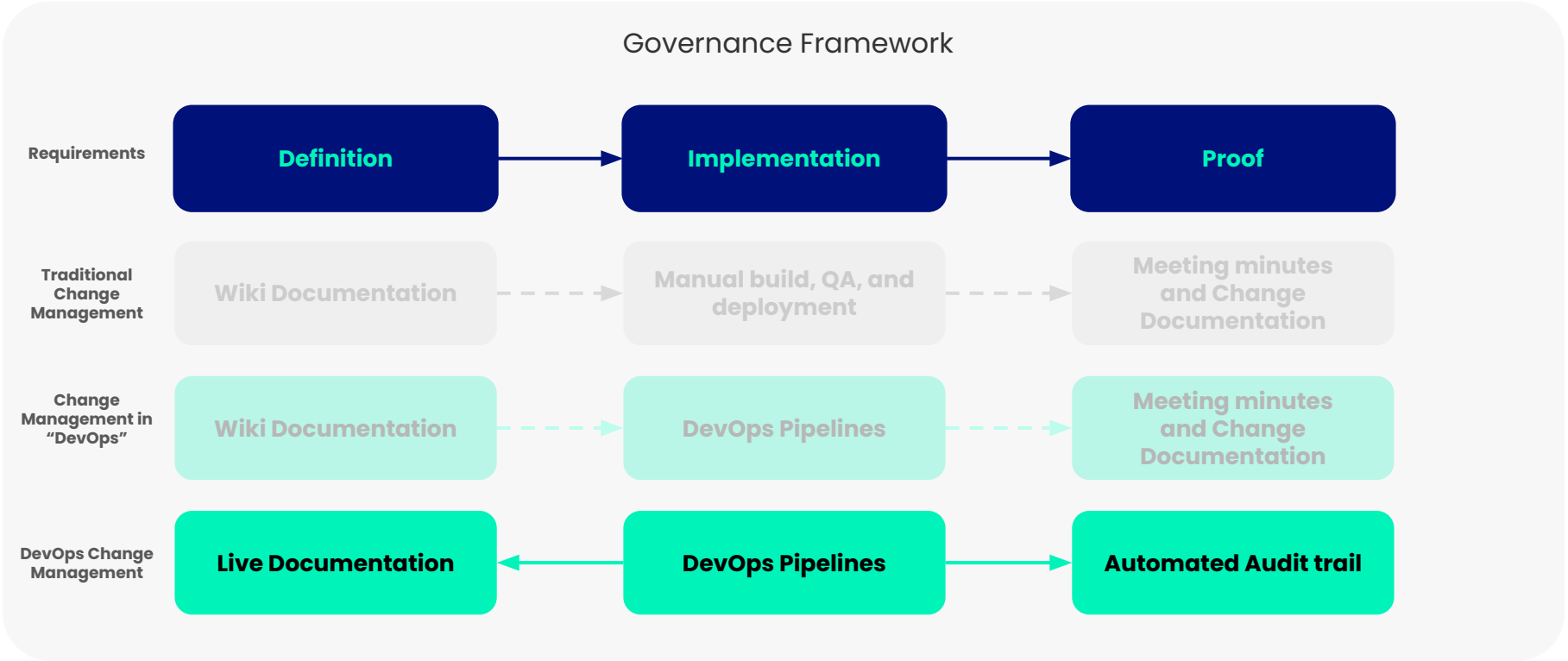
Continuous Change Management



Continuous compliance leads to incremental delivery, shorter lead times, and lower release risks.



Change Management Comparison



Implementing DevOps Change Management

merkely

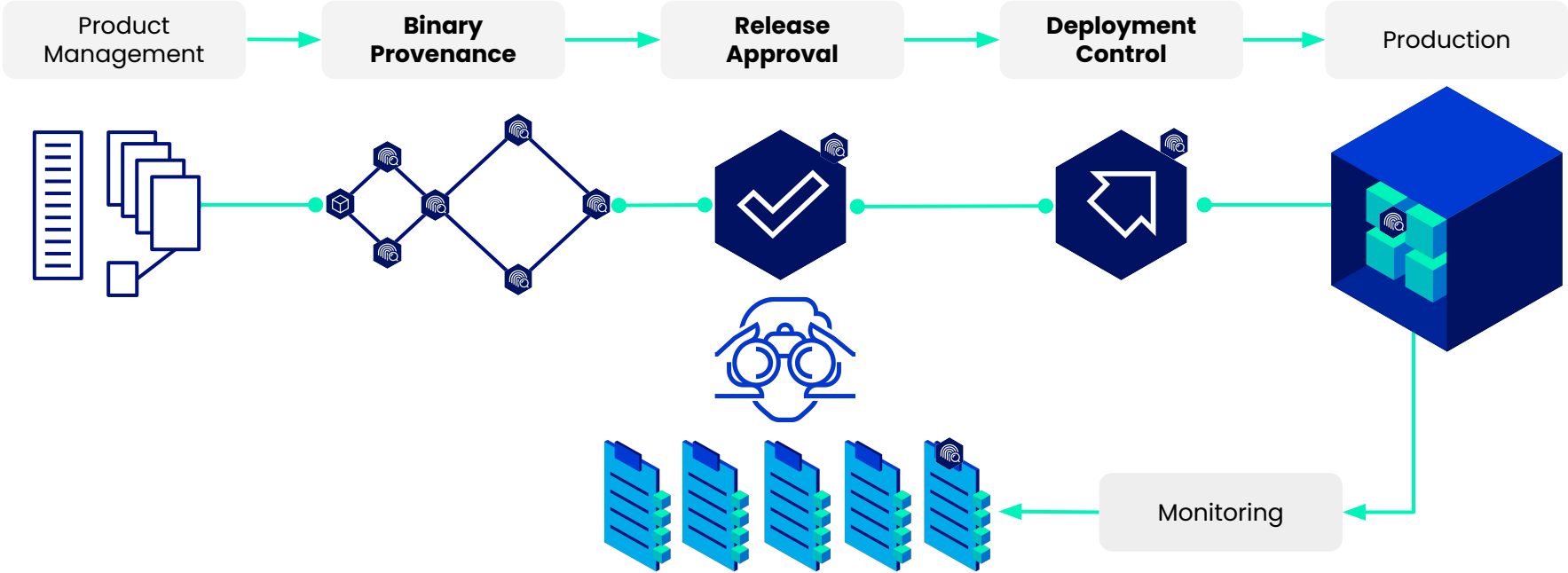
Insider Threat

> *"At Google, we want to do as much as we can to minimize the potential for Google personnel to use their organizational knowledge or access to user data in an unauthorized way—this includes running an unauthorized job"*
> https://cloud.google.com/security/binary-authorization-for-borg/

Google Cloud

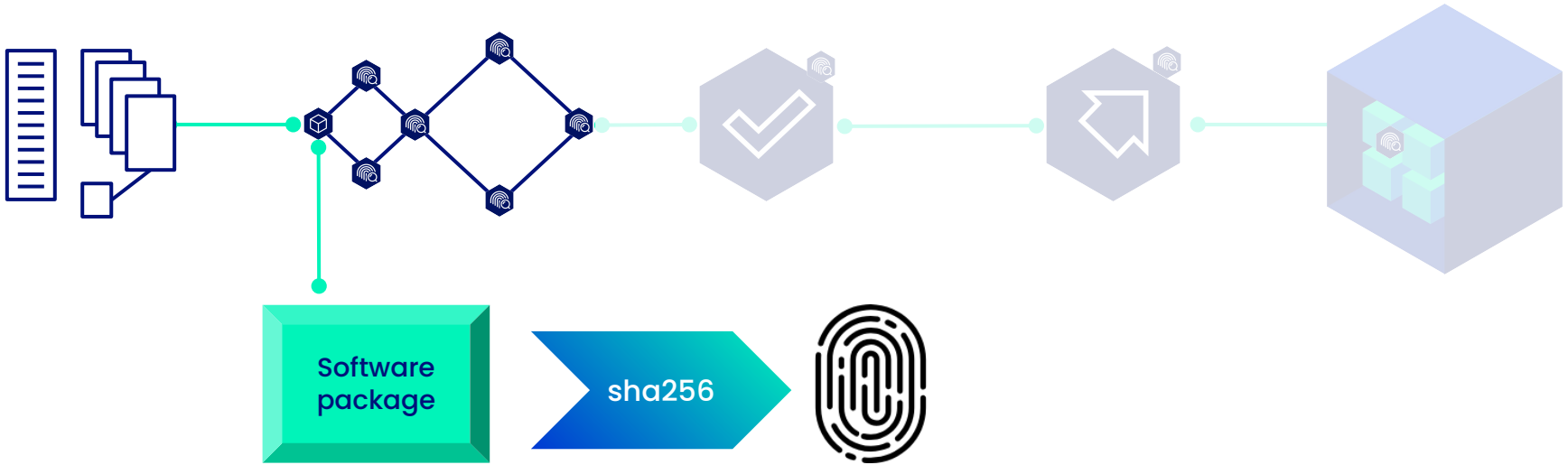
A Secure Chain of Custody across the value stream

Cryptographically ensure deployments are known, approved and compliant.

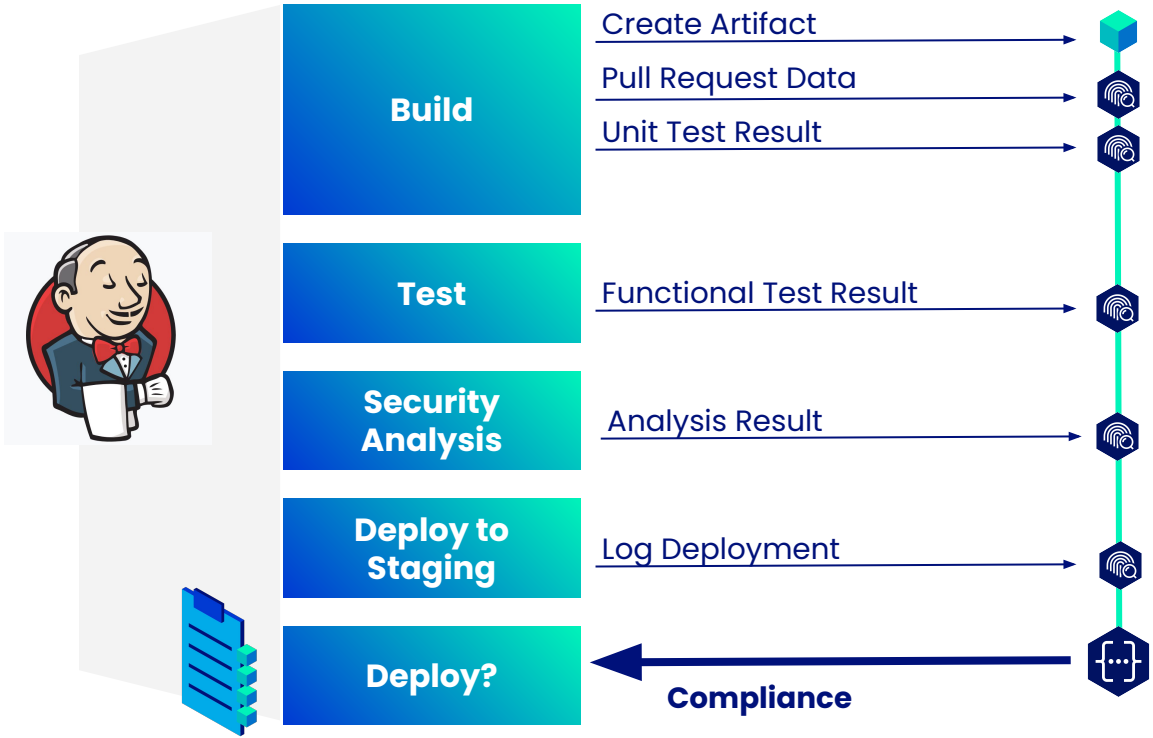


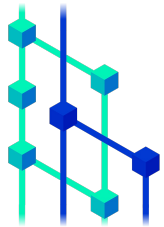
Binary Provenance

Fingerprints in a controlled build process provides a tamper-proof identity for all binaries across the value stream.



Control Evidence

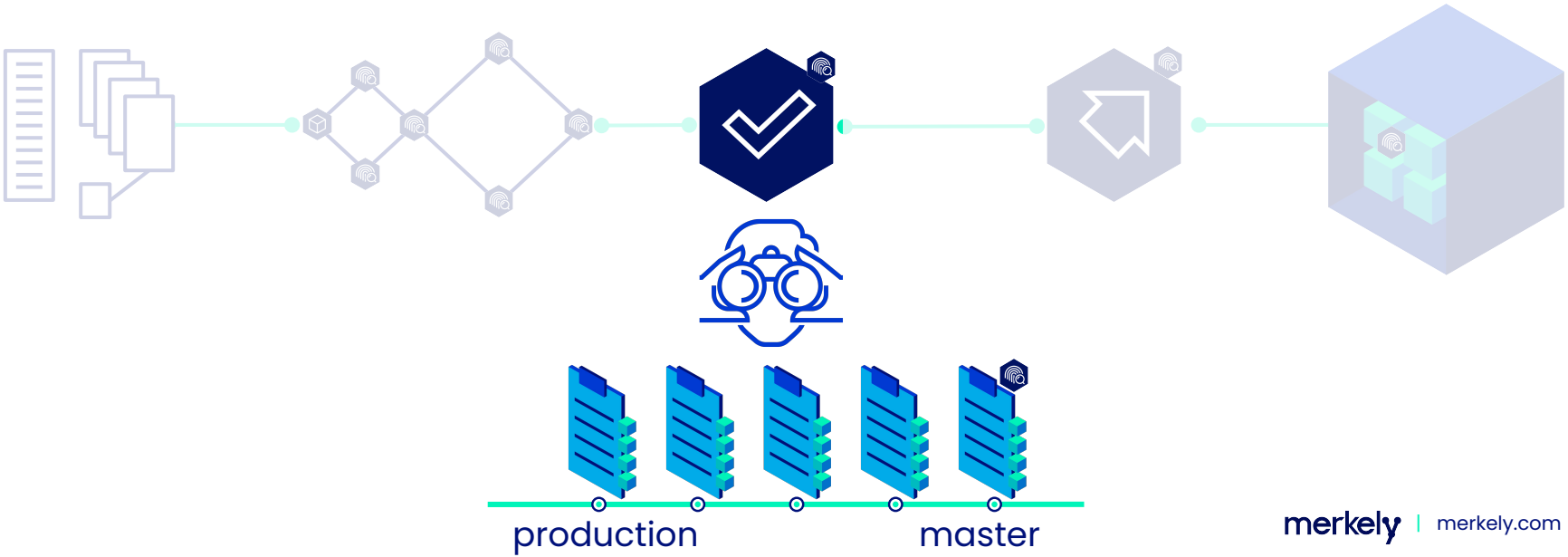




A **Secure Audit Trail** to prove compliance based on open standards

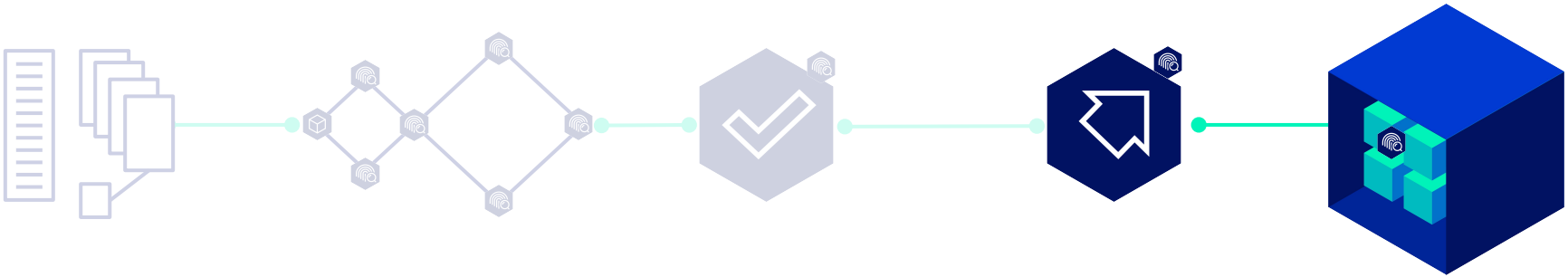
Documented Approvals

Automated change reports generated via version control or continuous integration events.



Automated Deployment Controls

Automatically ensure **only compliant software is deployed** by verifying binary provenance automatically as part of your deployment process.



Merkely delivers
Change Management at DevOps scale

- A Secure Chain of Custody
- Risk Controls as Code
- Provable audit trails
- Real time process documentation
- Empowered Teams

merkely

Mike Long
@meekrosoft
mike@merkely.com
+47 48676360