JAMF NATION LIVE

Compliance Benchmarks at Scale:

API-Driven Automation for Management and Reporting

Jan Voženílek

SENIOR PRODUCT OWNER

Richard Mallion

SENIOR EDUCATION SERVICES ENGINEER

Agenda

Introduction

Why Compliance Matters

Creating Benchmarks via API

The Power of Programmatic Creation

Under the Hood

What Happens Behind the Scenes

Compliance Reporting via API

Building Custom Reporting Solutions

Future & Roadmap

What's Coming Next

Why Compliance Matters

Dissecting the complex topic of compliance.

Why it matters more than ever?

What are consequences of non-compliance?

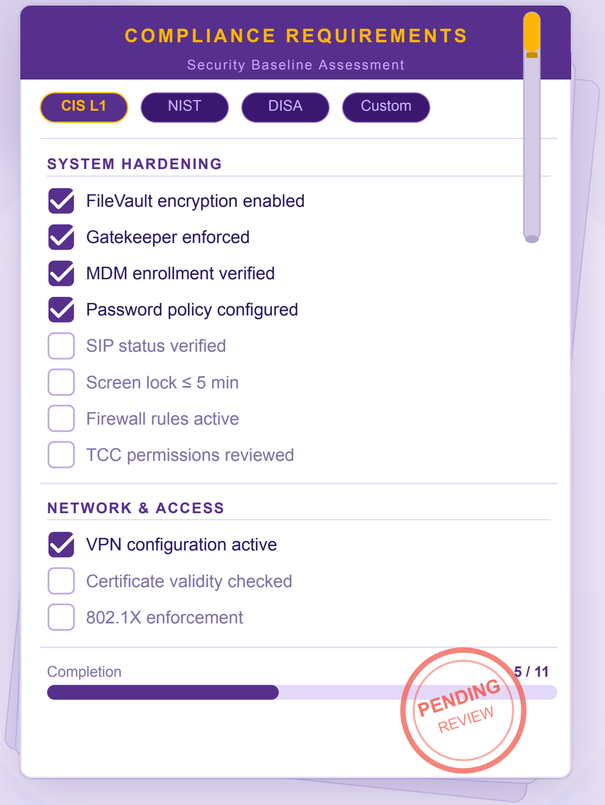
How to scale and automate it?

WHY COMPLIANCE MATTERS

Why implement **Baselines and Benchmarks?**

Some government agencies require all computers that interact with their systems and data to be subject to a specific benchmark with few allowances for exceptions. Many regulated industries will also be required to implement a security benchmark.

- IT and Information Security departments need to collaborate
- Balance between information security and user productivity
- Different devices with different risk categories



WHY COMPLIANCE MATTERS

Consequences of **Non-Compliance**

Monetary loss

- Fines
- Settlements
- Remediation costs

Loss of customer trust

- Damaged accounts
- Reputational harm

Public exposure

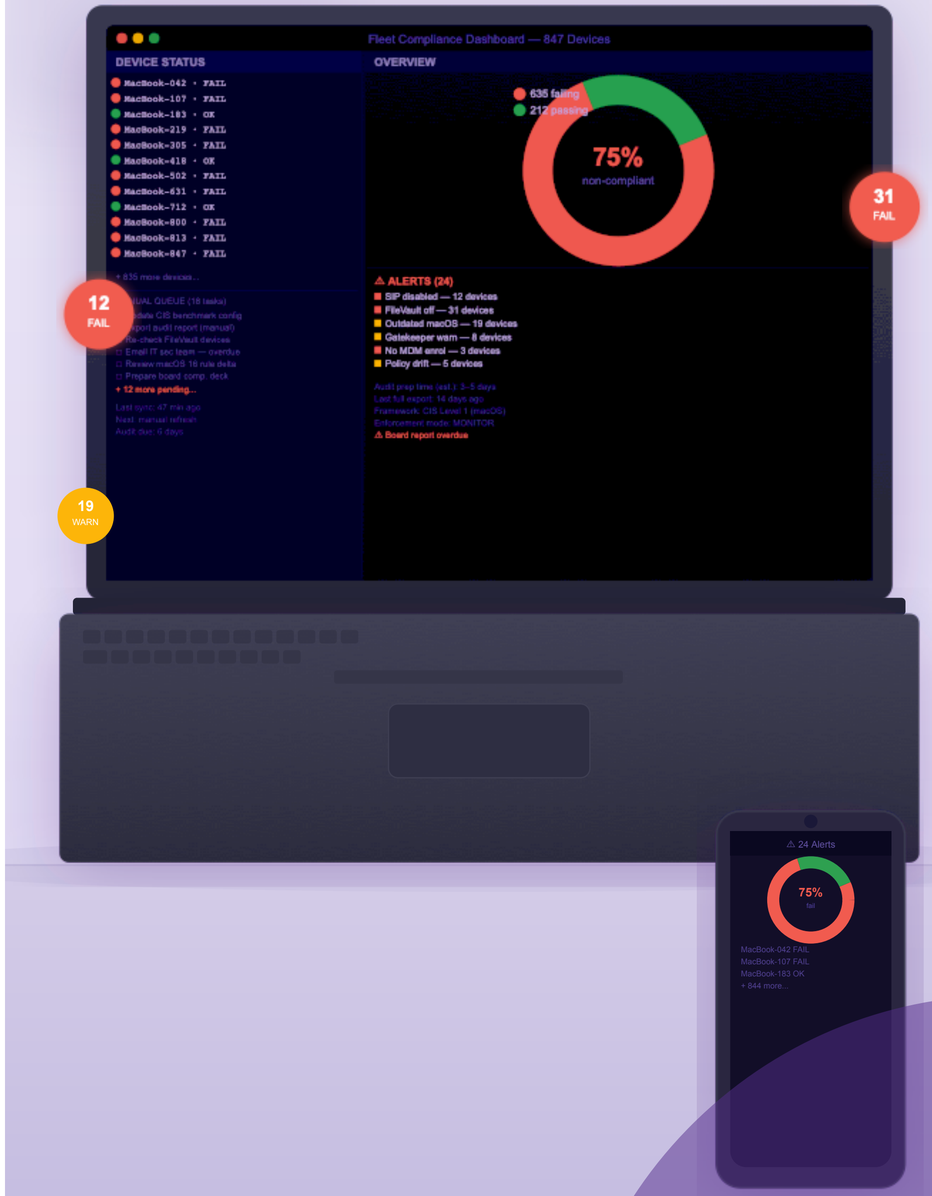
- Regulatory disclosures
- Press coverage

WHY COMPLIANCE MATTERS

Compliance at Scale Is Hard

- Managing benchmarks manually across 500+ devices is error-prone
- Each new macOS release can shift compliance requirements
- Multiple teams (IT, Security, GRC) need consistent, timely data
- Audit preparation takes days instead of minutes

The UI is great for getting started. APIs are how you scale.



WHY COMPLIANCE MATTERS

From Manual to **Automated**

Manual

- Spreadsheet workflows
- Ad-hoc scripts
- Inconsistent enforcement

Semi-automated

- UI-based benchmark deployment
- Manual updates
- Scheduled reports

Fully automated

- API-driven creation
- Programmatic reporting
- Integrated with SIEM/ticketing

Creating Benchmarks via API

The power of programmatic creation.

Why automate benchmark creation?

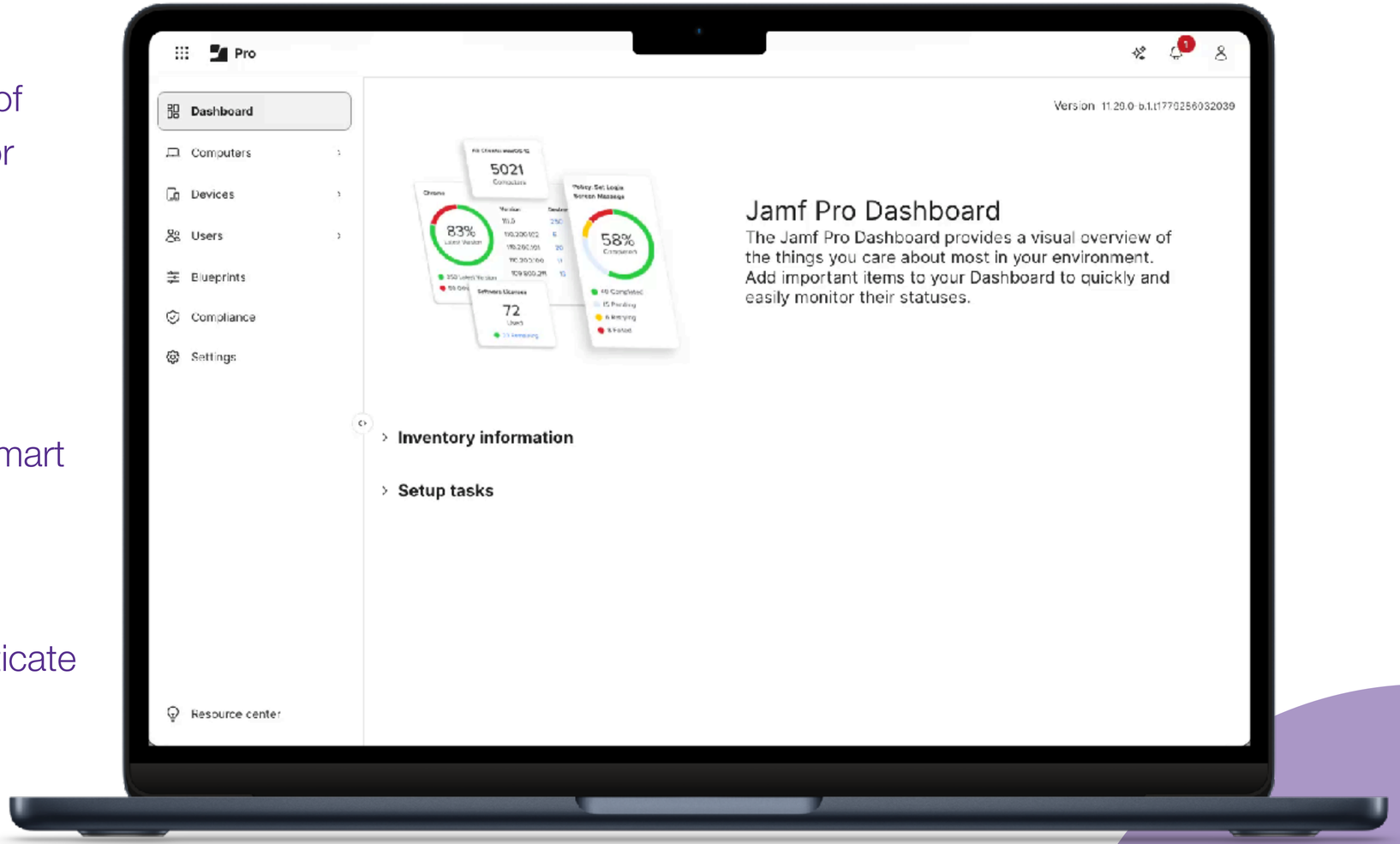
What the API payload looks like?

How to authenticate into Jamf Platform API?

CREATING BENCHMARKS VIA API

Quick recap: **Compliance UI** in Jamf Pro

- A compliance benchmark is a curated set of security rules (based on CIS, NIST, DISA, or custom) deployed to your Mac fleet
- Powered by mSCP (macOS Security Compliance Project) under the hood
- Built on top of existing Jamf Pro objects: smart groups, extension attributes, configuration profiles, scripts, categories, policies
- Requires Jamf Pro cloud + SSO to authenticate across Jamf Platform capabilities



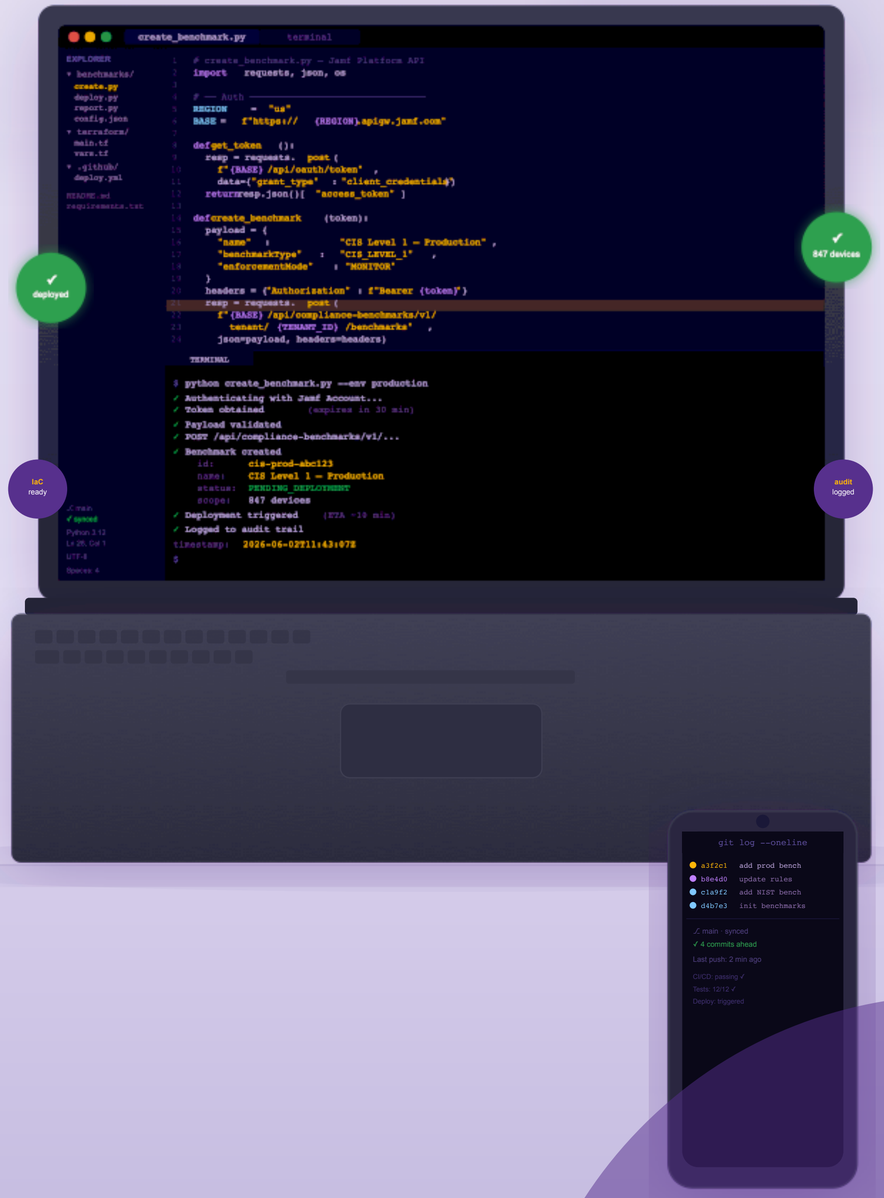
JAMF NATION LIVE

JNL LIVE DEMO

CREATING BENCHMARKS VIA API

Use Cases: Why Should I **Script Benchmark Creation?**

- **Standardization** — Identical configs across environments
- **Version control** — Benchmark definitions as JSON in Git; diff, review, roll back; **IaC in Terraform**
- **Multi-tenant / MSP** — One script deploys the same baseline across dozens of customer instances
- **CI/CD integration** — Benchmark deployment as part of environment provisioning
- **Audit readiness** — Timestamped record of every deployment, automatically



Under the Hood

Architecture overview.

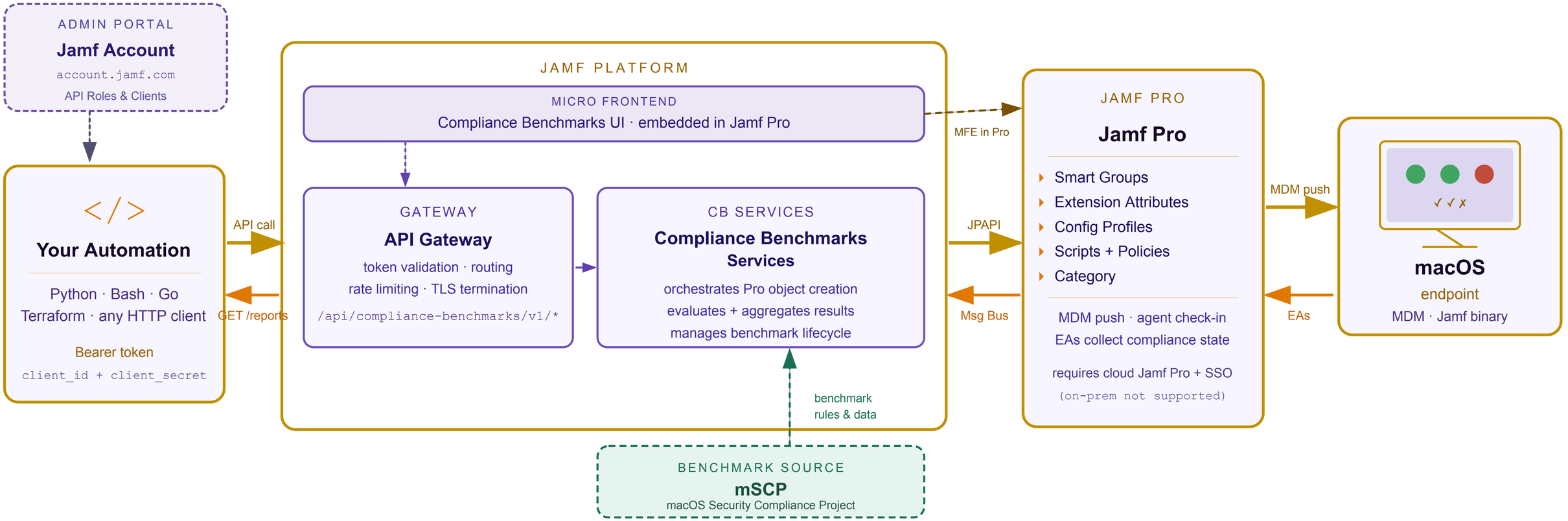
Why Jamf Platform is the key element?

What happens behind the scenes?

How rules are evaluated?

UNDER THE HOOD

Jamf Platform ↔ Jamf Pro: How It Connects

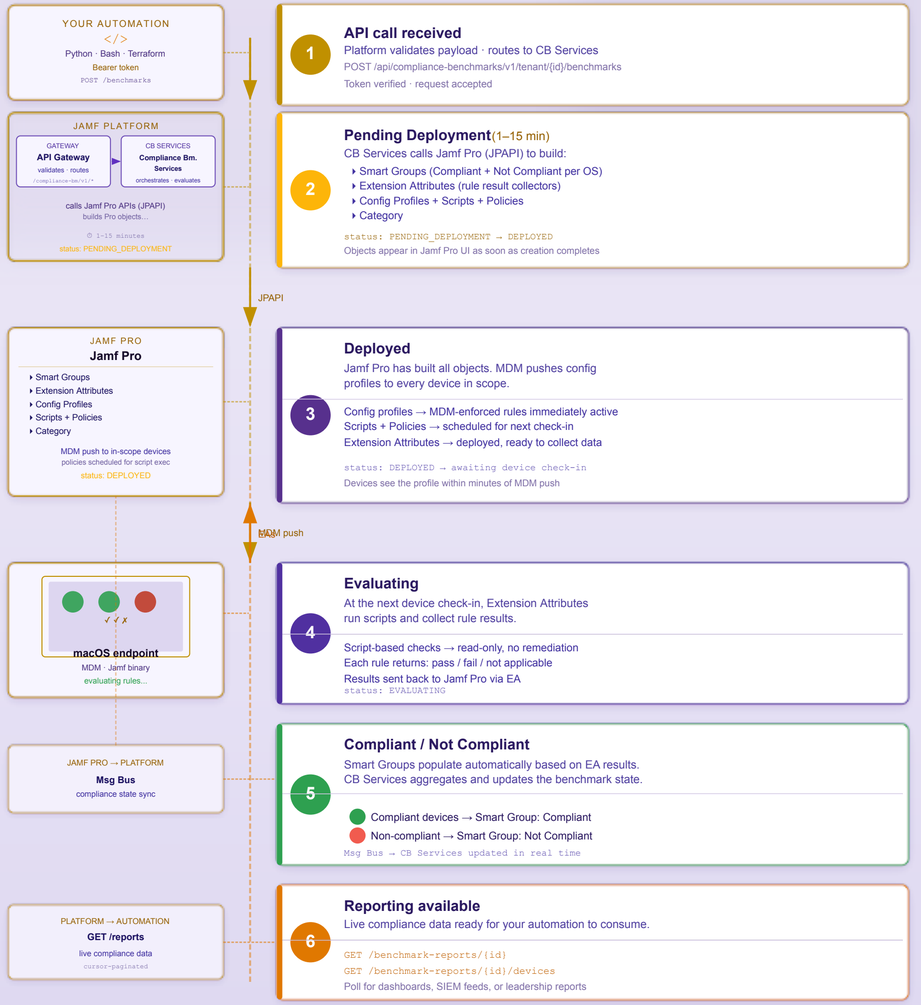


Deployment / API call · Compliance reporting · Admin setup / config · MFE embedded in Pro · Benchmark source data

UNDER THE HOOD

Benchmark Lifecycle: From API Call to **Device State**

- **API Call received** → Platform service validates payload, calls JPAPIs
- **Pending Deployment** (1–15 min) → Jamf Pro builds objects: smart groups, configuration profiles, scripts, policies, extension attribute, category
- **Deployed** → MDM pushes configuration profiles to in-scope devices; policies scheduled for script execution
- **Evaluating** → Extension attribute collects rule results at next check-in
- **Compliant / Not Compliant** → Smart groups populate; reporting reflects current state
- **Reporting available** → API reporting endpoints return live compliance



UNDER THE HOOD

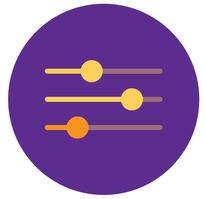
On the Device: **Evaluation and Enforcement**

Evaluation (Monitoring)

- Rules checked, results reported back
  - Script-based check (read only)
  - Via policy and extension attribute
- No remediation

Enforcement (Remediation)

- Configuration profile rules
  - MDM-enforced
  - No user override
- Script-based settings
  - Policy triggered script execution
  - Targeting non-compliant devices



Configuration

- Tailor out any baseline
  - Start with Monitor, promote specific rules to Enforce as you gain confidence
- ODVs (Organizational Defined Values)
  - Configurable thresholds, e.g. password length minimums

Building Custom Reporting Solutions

Compliance reporting via API.

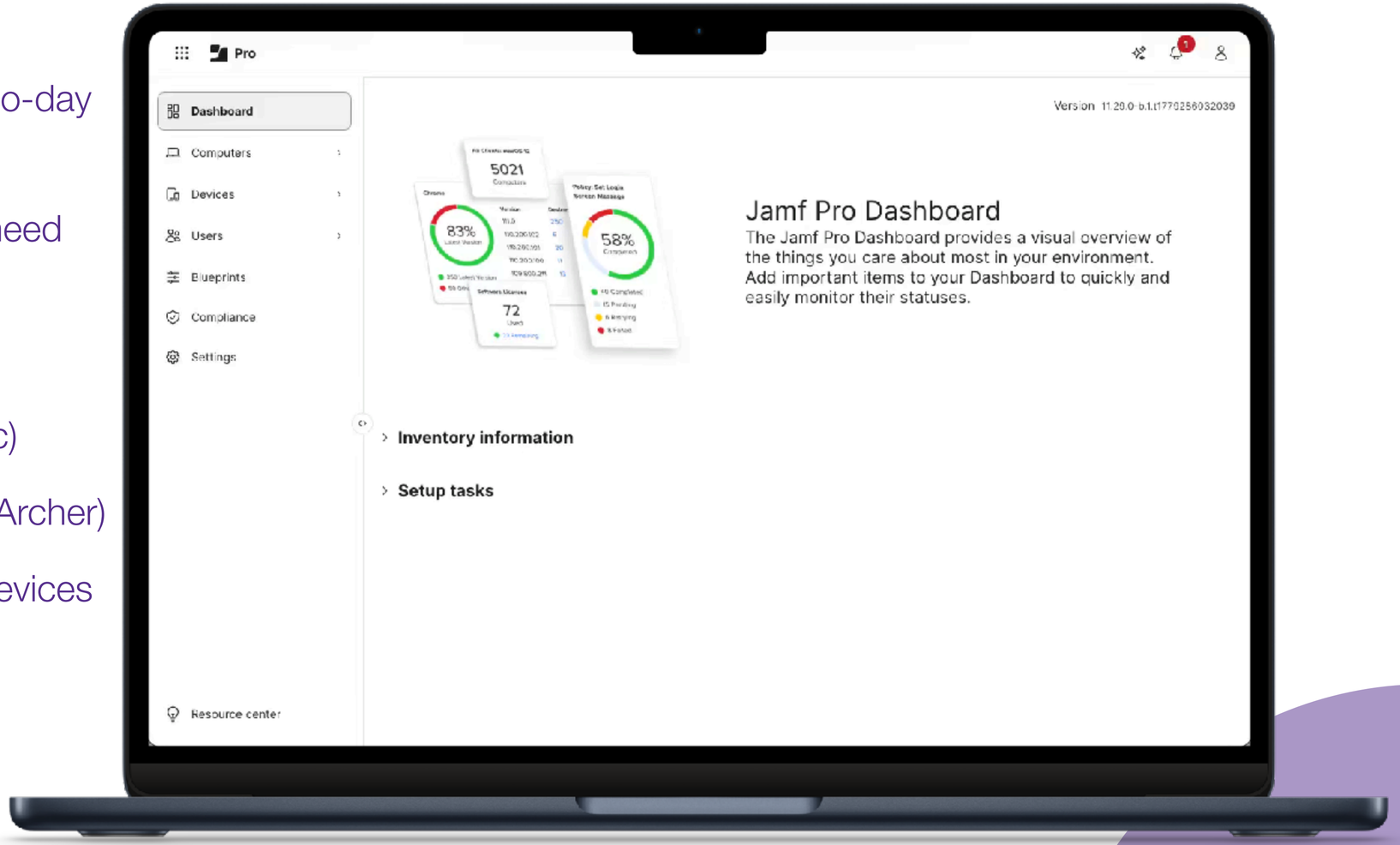
Why go beyond the built-in dashboard?

What does the reporting API return?

How do you integrate compliance data with your existing tools?

Quick recap: Built-in **Rules report**

- Jamf Pro's compliance UI is great for day-to-day monitoring
- But Security, GRC, and leadership teams need compliance data in *their* tools
- Common needs:
  - Export to SIEM (Splunk, Sentinel, Elastic)
  - Feed into GRC platforms (ServiceNow, Archer)
  - Auto-create tickets for non-compliant devices
  - Weekly compliance trend reports for leadership

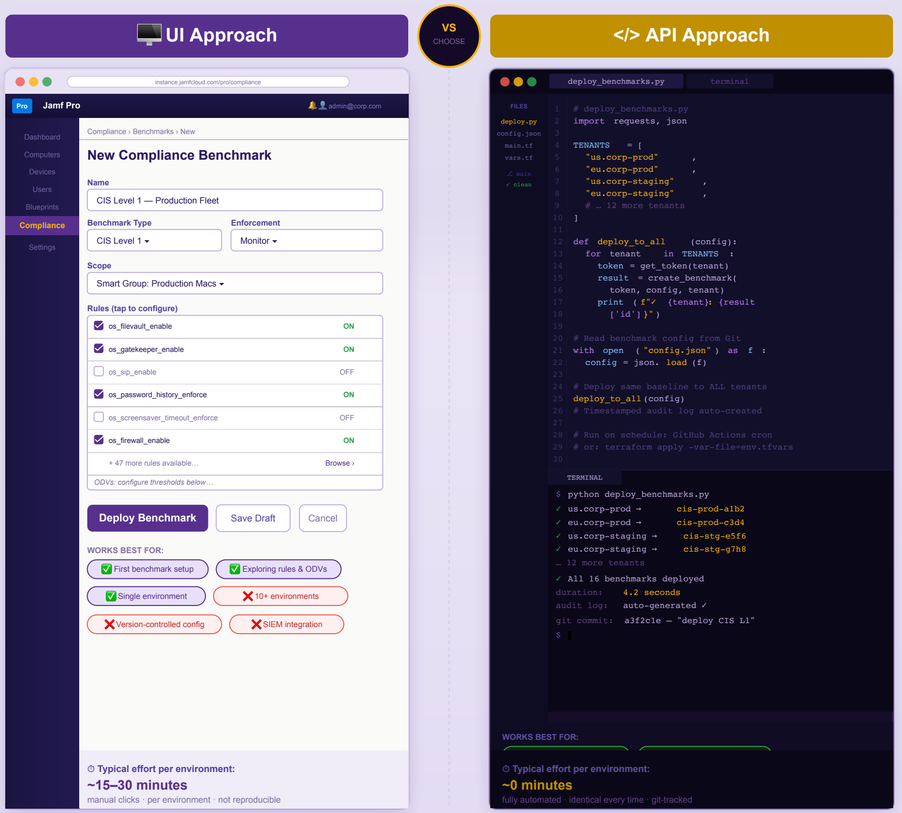


JNL LIVE DEMO

BUILDING CUSTOM REPORTING SOLUTIONS

Choosing **Your Approach**

Scenario	UI	API
First benchmark setup	✅ Best	Works
Exploring rules and ODVs	✅ Best	Possible
Single environment	✅ Best	Overkill
10+ environments	❌ Slow	✅ Best
Version-controlled config	❌ No	✅ Best
Custom reporting	❌ Limited	✅ Best
SIEM / ticketing integration	❌ No	✅ Best
Day-to-day monitoring	✅ Great	Works



What is Coming Next

Future & Roadmap.

Why we build X and not Y before Z?

What's on the roadmap?

How can you influence what we build?

WHAT IS COMING NEXT

On the **Roadmap** (as of today)

Now

- Scope to multiple smart and static computer groups
- Select OS version(s) to cover
- Continuously add new baselines from mSCP GitHub

Next

- Adopt mSCP 2.0
- Support macOS 27
- Edit benchmark via API
- Any small tweaks based on feedback

Later

- Compliance for iOS/iPadOS
- Compliance in Jamf School
- Management via Blueprints
- Full compatibility with DDM

WHAT IS COMING NEXT

We Need **Your Input**

- Every feature shown today started as community feedback
- What would make compliance automation work better for your org?
- Join the conversation:
  - Jamf Nation (*community.jamf.com*)
  - Feature requests (*ideas.jamf.com* or via your Jamf account team)
  - MacAdmins Slack (*#jamf-compliance-benchmarks*)
  - Talk to us after this session

WHAT IS COMING NEXT

What to **Take Home**

The APIs unlock scale

developer.jamf.com/platform-api/

- Automate benchmark deployment
- Eliminate manual repetition
- Enforce consistency

Reporting is where the value compounds

- Pull compliance data into your SIEM, GRC, or ticketing system
- Don't make your CISO screenshot Jamf Pro

Start small, automate incrementally

- Pick one workflow today and build from there

Thank You!

Questions?

JAMF.IT/JNL2026CBAPI

Jan Voženílek

SENIOR PRODUCT OWNER

Richard Mallion

SENIOR EDUCATION SERVICES ENGINEER