AIRLOCK®

Airlock Partner-Tech-Community Online training

Episode 6 – May 2026

Passkeys - Tracing - Failover & Loadbalancing

Agenda

5min	Welcome	Stefan Braun
25min	Passkeys	Urs Zurbuchen
25min	Tracing	Stefan Braun
25min	Failover and Loadbalancing	Friedrich Oesch
10min	Q&A	

Total: 90min

AIRLOCK

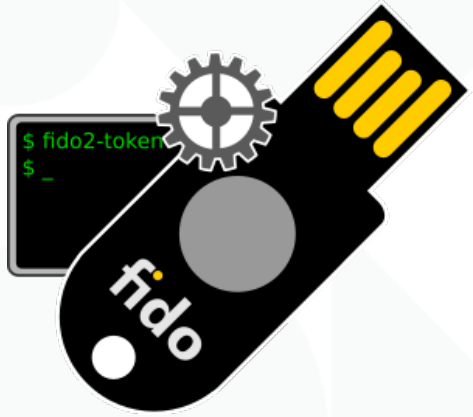
PASSKEYS

The modern way to sign in.
Secure. Simple. Phishing-resistant.

Passkeys and FIDO – Some History

- Goal of FIDO
  - Reduce over-reliance on passwords
  - Phishing protection

- Issues with FIDO
  - Strict device binding
  - No recovery
  - Low integration, high friction



Passkeys

Advantages

- Based on FIDO2
- Easier to use: native OS integration
- Cloud synchronisation

Disadvantages

- Cloud synchronisation
- Are they really secure?

Impact of US CLOUD Act

- Access to private key
- End-to-end encrypted
- Zero-knowledge

Recovery ?

Impact of US CLOUD Act

- MacOS, iOS
  - Protected only if Advanced Data Protection is enabled
  - Recovery: 28 character code or trusted contact

UNCONTROLLABLE

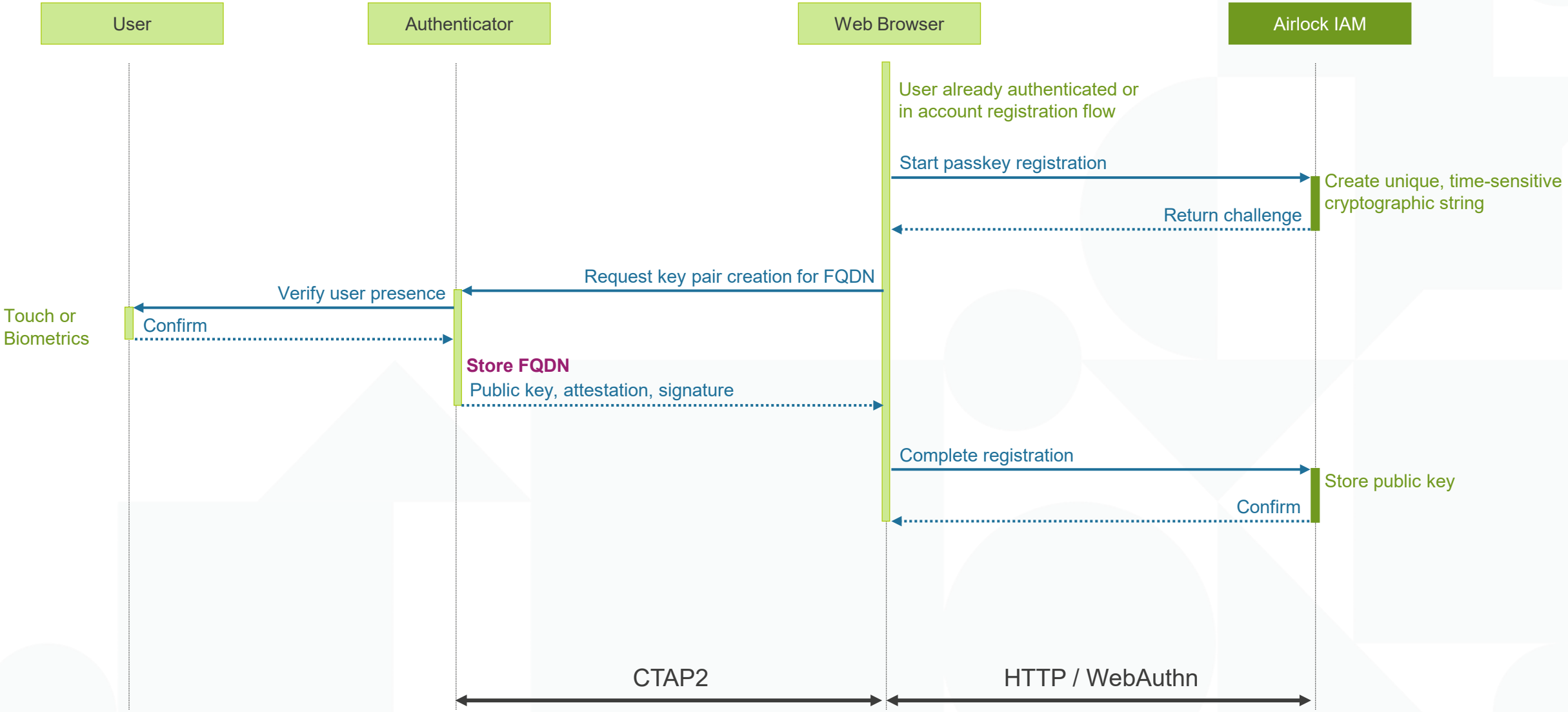
- Android
  - Google Password Manager with sync
  - Zero-knowledge enabled by default
  - Recovery: lock screen pin/password manager pin

UNCONTROLLABLE

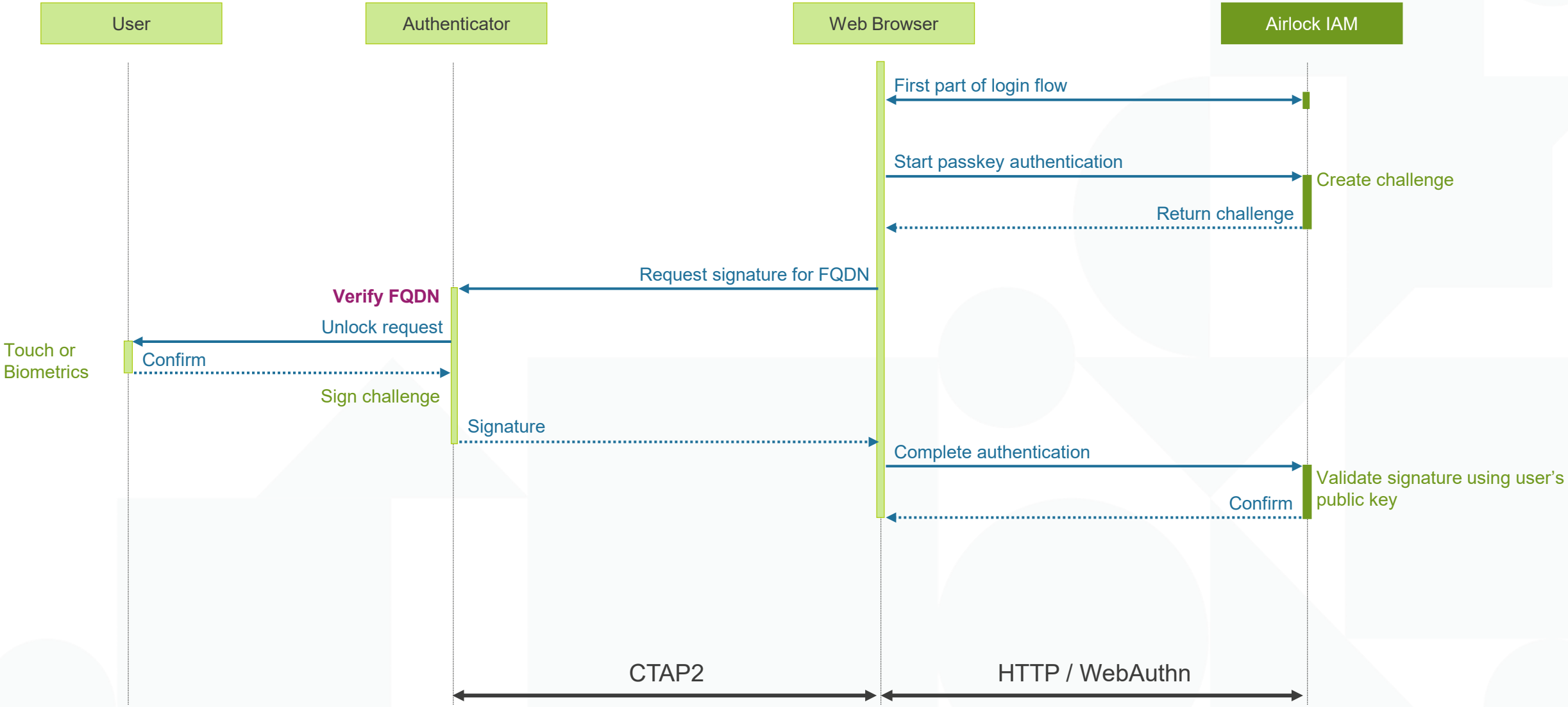
Impact of US CLOUD Act

- Windows
  - Microsoft Authenticator: Mobile app, supports only Microsoft services
  - Windows Hello: does not support sync'ing
  - Alternative Password Manager
  - Zero-knowledge: depends on password manager
  - Recovery: depends on password manager

Phishing Protection – Registration Flow

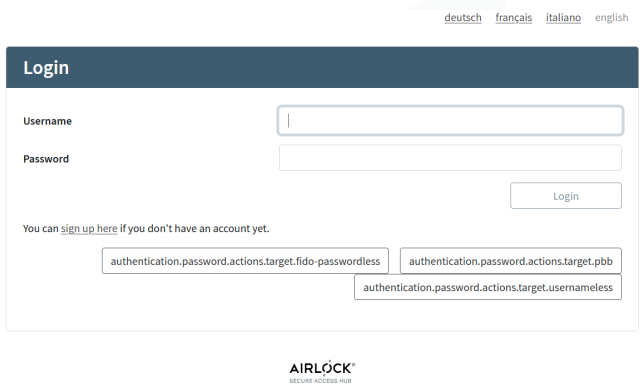


Phishing Protection – Authentication Flow



UI Considerations

User | Authenticator | Web Browser | Airlock IAM



No control over UX

- Built-in UIs (OS, Browser)
- Standard, unchangeable messages
- Un-stylable

No control

- User-selected
- Usually intuitive

Complete control over UX

- Texts
- Layout, styling

Rollout and User Enablement

UI-Break often leads to costly **helpdesk** calls

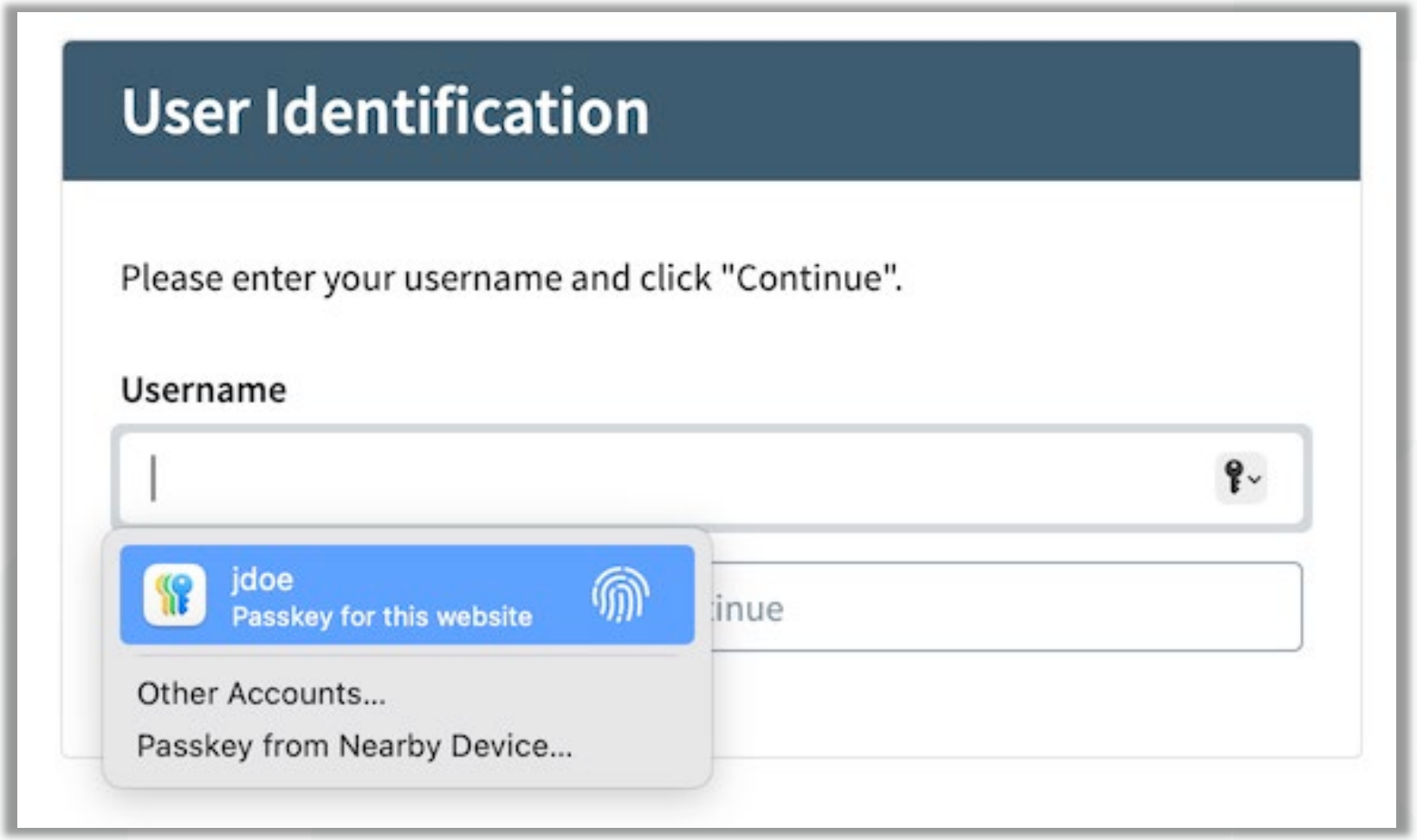
Solution 1
- No passkeys
- Remain exposed to phishing attacks

Solution 2
- Make it opt-in (with self-service)
- If user has passkey registered -> require it
- No inadvertent phishing
- Less support issues
- Very easy to do with Airlock Flows

Are Passkeys 2FA?

+

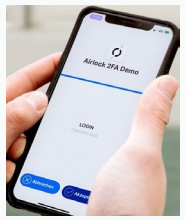
Are Passkeys 2FA?



Are Passkeys 2FA?

+

+



Passkeys and PSD2 (SCA)

- Generally not considered fully PSD2-compliant on their own
- PSD2 requires proof of possession of a specific, trusted device
  - Sync'ed passkeys are automatically backed up and copied across all your devices (e.g., via iCloud or Google Password Manager), it becomes impossible to guarantee that the user is authenticating from a specific, authorized physical device.
  - The private cryptographic keys are stored and transferred through third-party cloud platforms. Regulators have difficulty verifying that these cloud environments independently meet stringent customer authentication and encryption standards.

Passkeys and PSD2 (SCA)

PSD2

FIDO2

Device-bound FIDO2 keys are PSD2-compliant

Attestation

- Cryptographically verify
  - Make
  - Model
  - Authenticity

- Requires device-bound FIDO2 key

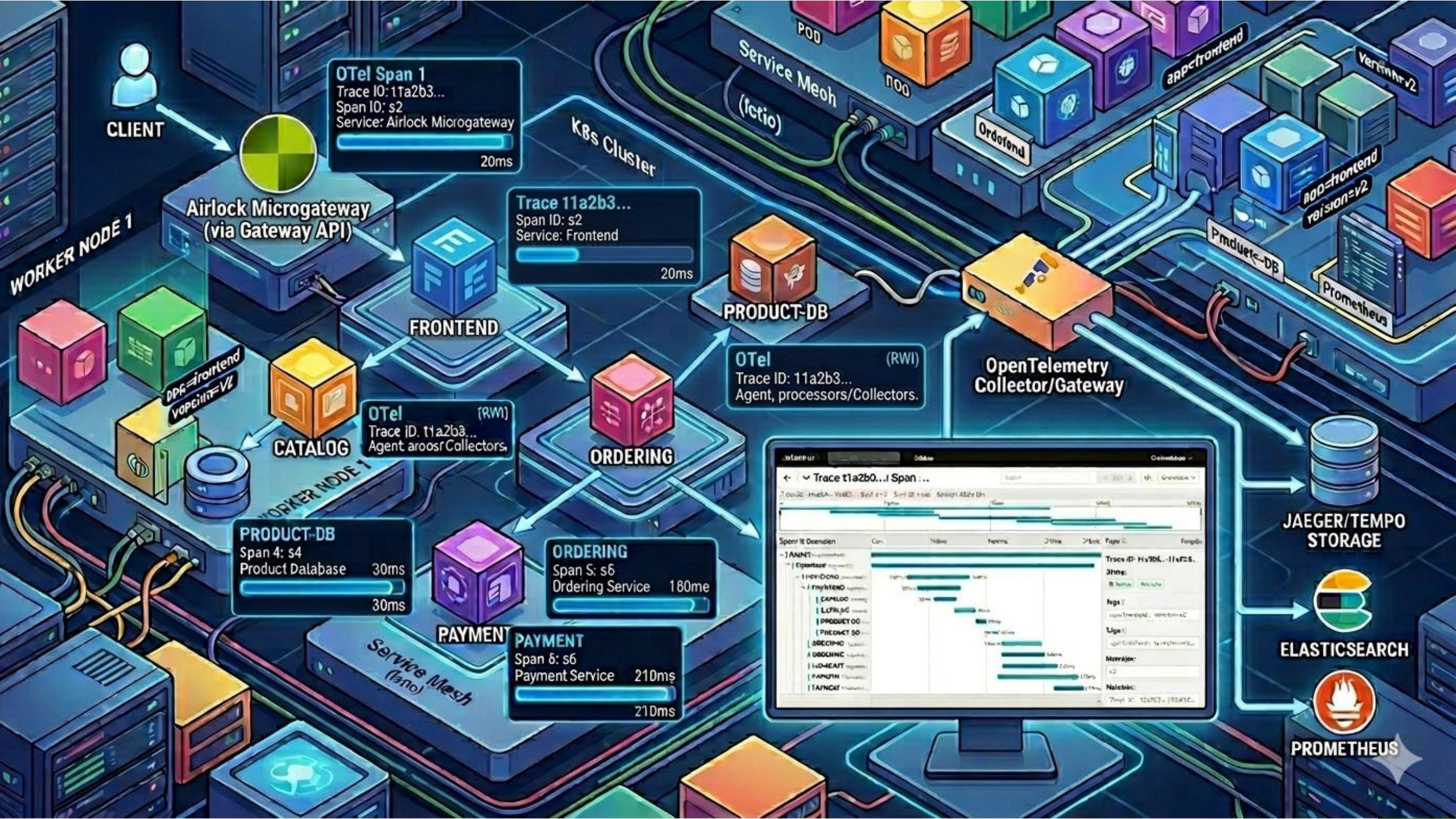
AIRLOCK

Tracing

Observe. Understand. Optimize.

The Core Problem

- **Distributed Blindness:** Logs often don’t tell the whole story of a request. With **tracing**, you can **follow** a **request end-to-end** and see **sequence** and **timing** of each **component involved**.
- **The "Shoebox" Dilemma:** Standard logs are disconnected receipts; searching them requires matching timestamps manually across multiple service logs.
- **The Blame Game:** High MTTR (Mean Time to Resolution) because teams don't know *which* service is slow.

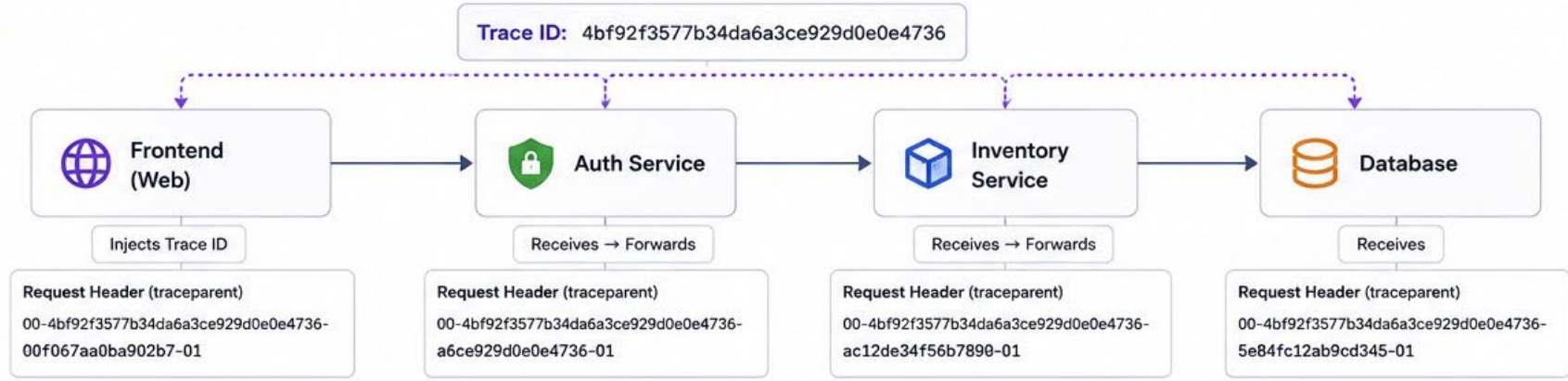

CLIENT
OTel Span 1
Trace ID: 11a2b3...
Span ID: s2
Service: Airlock Microgateway
20ms
Airlock Microgateway
(via Gateway API)
WORKER NODE 1
K8s Cluster
Service Mesh
(Istio)
Trace 11a2b3...
Span ID: s2
Service: Frontend
20ms
FRONTEND
PRODUCT-DB
OTel (RWI)
Trace ID: 11a2b3...
Agent, processors/Collectors.
OpenTelemetry
Collector/Gateway
CATALOG
OTel (RWI)
Trace ID: 11a2b3...
Agent, processors/Collectors.
ORDERING
PRODUCT-DB
Span 4: s4
Product Database 30ms
30ms
PAYMENT
ORDERING
Span 5: s6
Ordering Service 180ms
PAYMENT
Span 6: s6
Payment Service 210ms
210ms
JAEGER/TEMPO
STORAGE
ELASTICSEARCH
PROMETHEUS

The "Magic" of OTel

1 The Global Trace ID:
OTel injects a unique cryptographic ID into the metadata headers (like traceparent via W3C standards) of every network request.

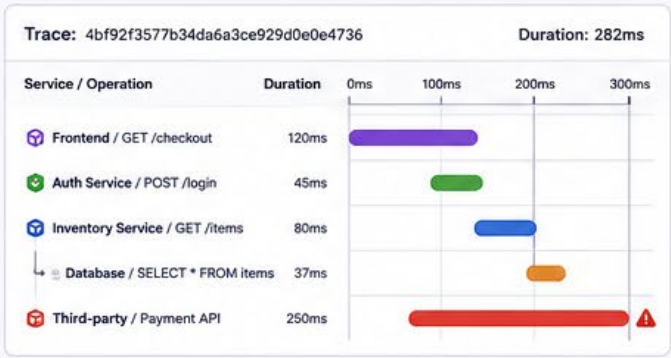
2 The Journey Map:
As a request hops from Frontend --> Auth Service --> Inventory --> Database, the ID travels with it.

3 Child Spans:
Every single operation becomes a sub-segment (span) of the master trace.



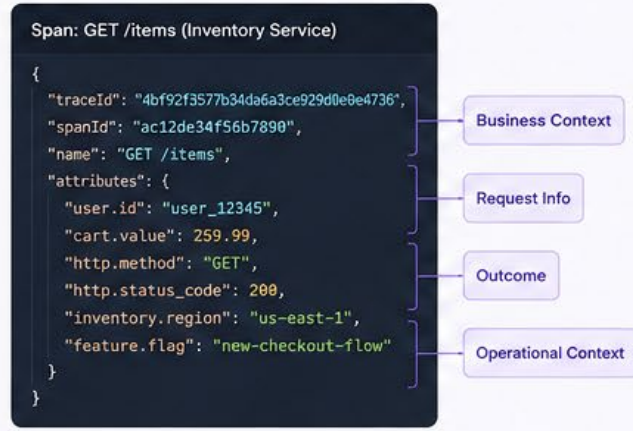
Real-Time Value

1 Instant Bottleneck Identification:
Visually isolate database query latencies vs. third-party API lag.



⚠ Bottleneck detected: Third-party / Payment API (250ms)

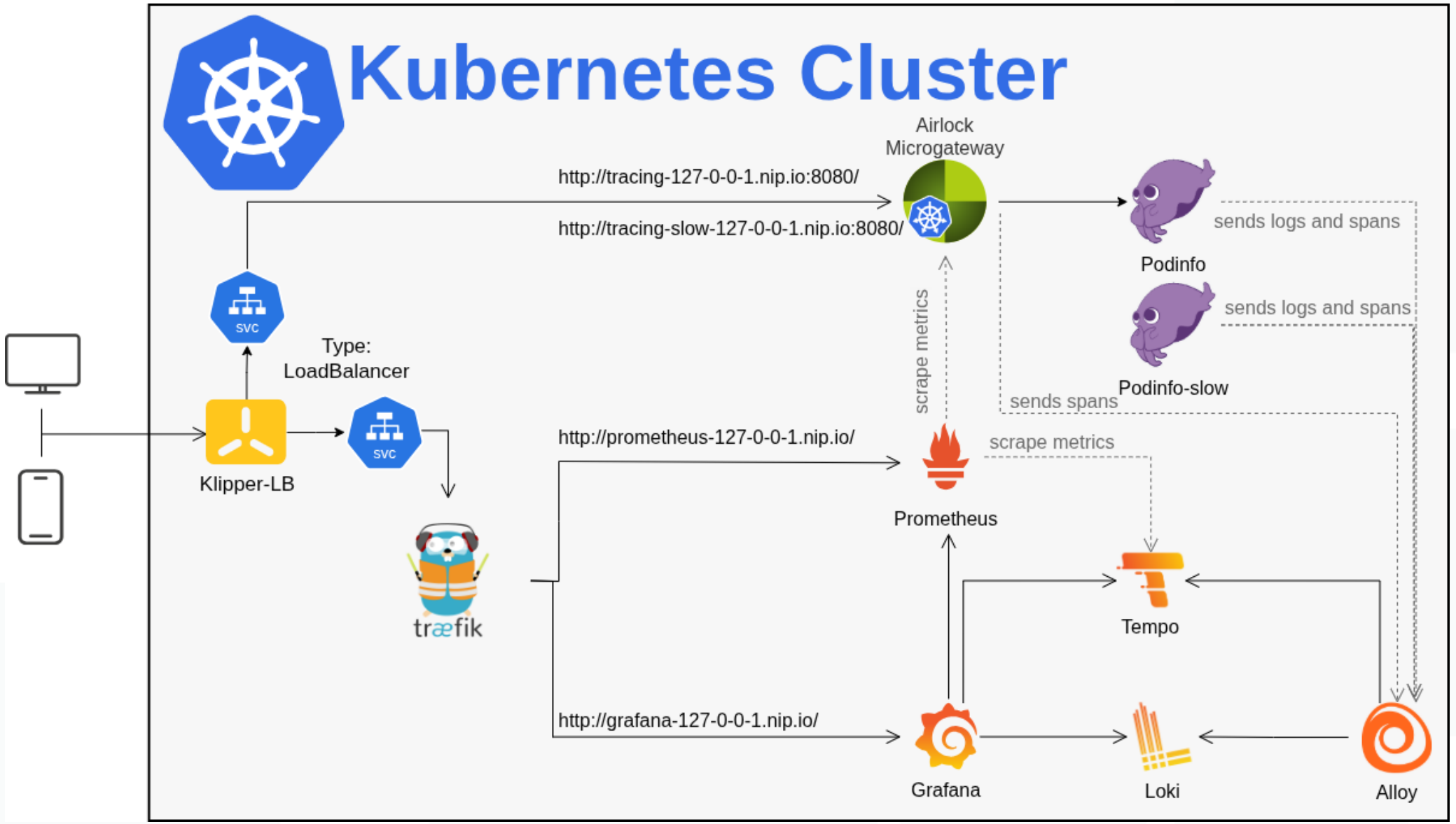
2 High-Cardinality Context:
Attach business metadata directly to spans (e.g., user.id, cart.value, http.status_code).



3 Correlation over Isolation:
Instantly jump from a broken trace directly to the exact application logs generated by that specific thread.

One click from trace → exact logs for that thread

Topology


Kubernetes Cluster
Airlock
Microgateway
http://tracing-127-0-0-1.nip.io:8080/
http://tracing-slow-127-0-0-1.nip.io:8080/
Podinfo
sends logs and spans
Podinfo-slow
sends logs and spans
scrape metrics
sends spans
SVC
Type:
LoadBalancer
Klipper-LB
SVC
http://prometheus-127-0-0-1.nip.io/
scrape metrics
Prometheus
træfik
Tempo
http://grafana-127-0-0-1.nip.io/
Grafana
Loki
Alloy

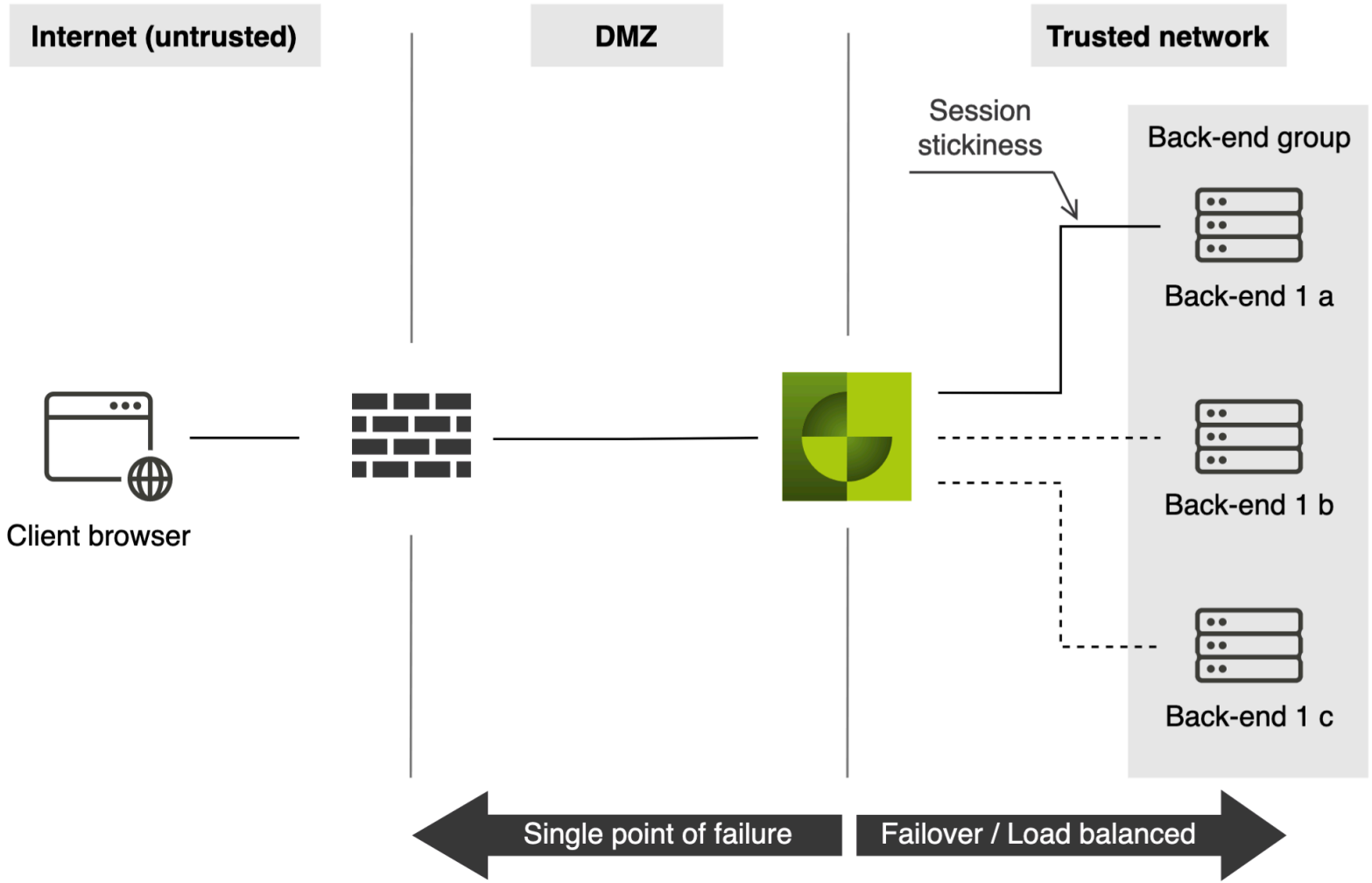

airlockctl assss
DETECTED!
LIVE DEMO

AIRLOCK

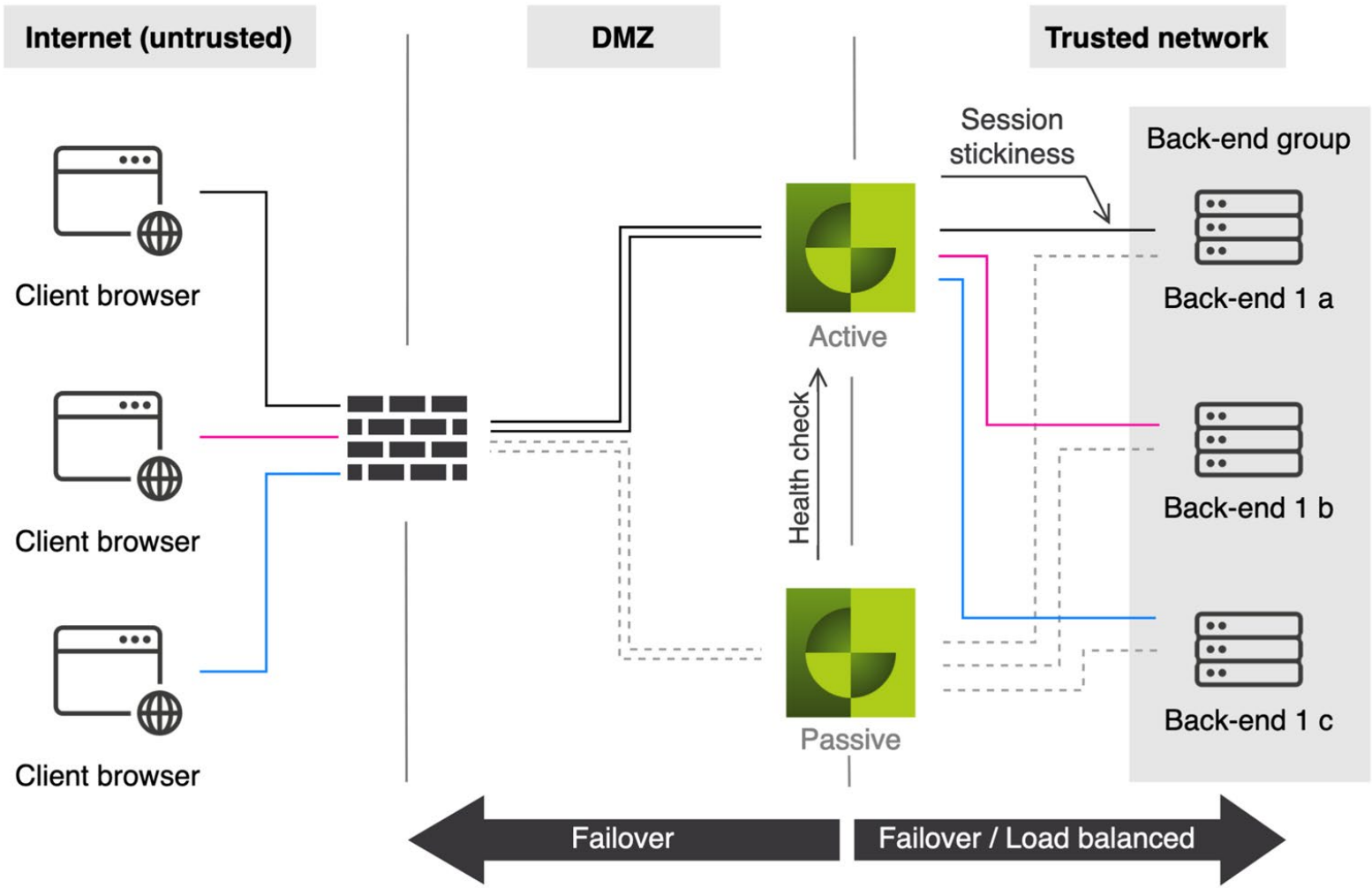
FAILOVER & LOADBALANCING

Stay up. Stay safe. Stay in business.

Single Airlock Gateway



Failover Cluster



Failover Cluster

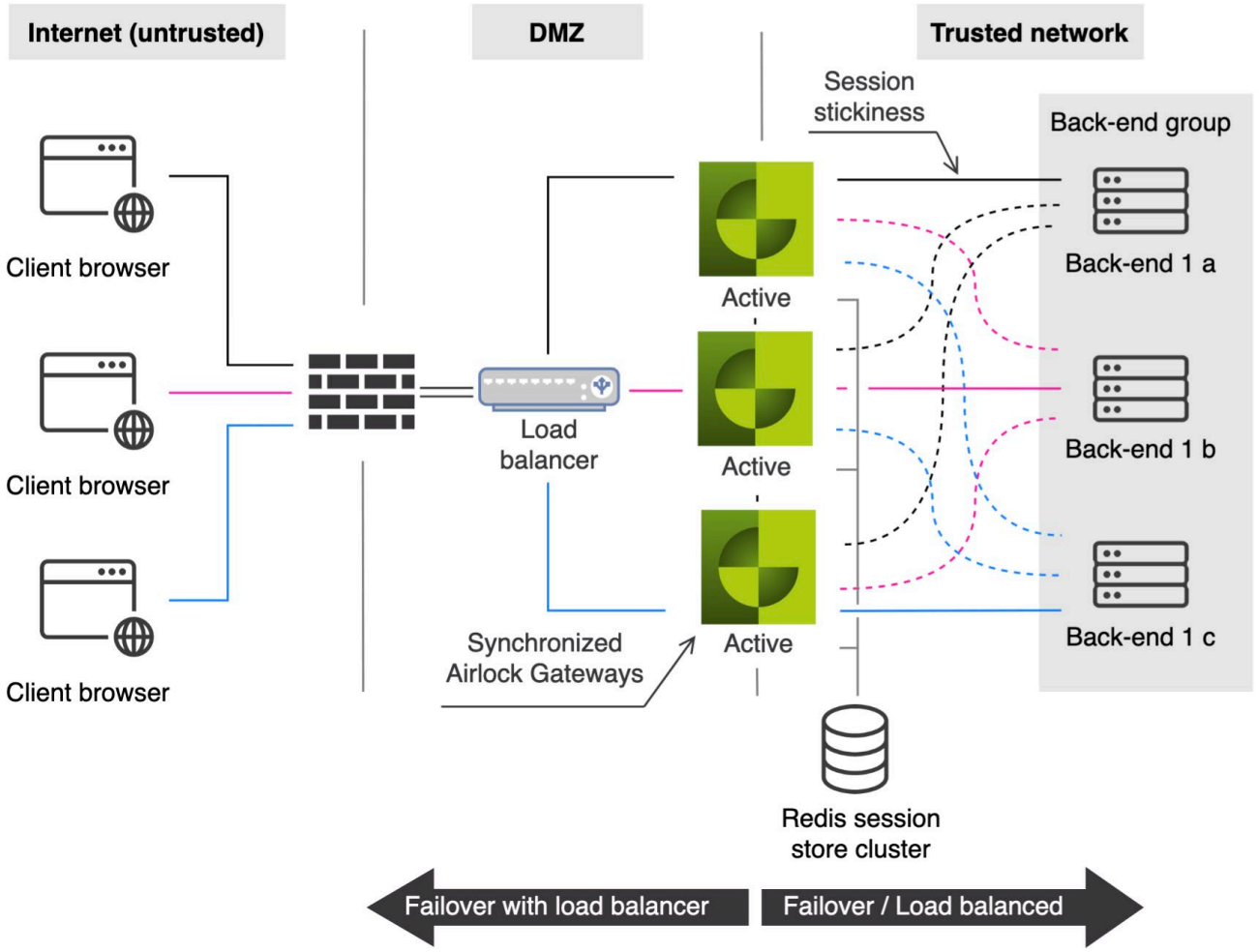
+ Cost-sensitive setup
+ Recommended for production environments
+ Easy to set up and maintain
+ High availability through failover instance
+ Availability even during update processes
+ Mutual monitoring of both instances
+ Health checks by the passive Airlock Gateway
+ Internal process monitoring with self-healing

- Only one system is active at a time

Failover – What to know

- There is no primary!
- Don’t use different versions in one cluster!
- For rollback reasons set «old» cluster member to offline!
- Renegotiation after "Active - Active" has priority on last processing cluster member.

Load Balanced Airlock Gateway Farm (with external Redis Store)



an external Redis Store can keep the session states and stickiness in sync for all gateways

Load Balanced Airlock Gateway Farm (with external Redis Store)

+ Works with two or more Airlock Gateway instances
+ Recommended for production environments if load exceeds capacity of a single instance
+ Scalable setup for growing throughput and availability demands
+ All systems can be active at a time
+ Availability even during the update process is possible
+ Application-related load balancing
+ Health checks are managed by the load balancer
+ Mixed hardware can be used for the Airlock Gateway instances
+ Internal process monitoring with self-healing
+ Sessions are synchronized between the Airlock Gateway instances

- Higher setup costs
- Requires a dedicated load balancer
- Requires one or more Redis session stores
- Can be more complex to set up and maintain

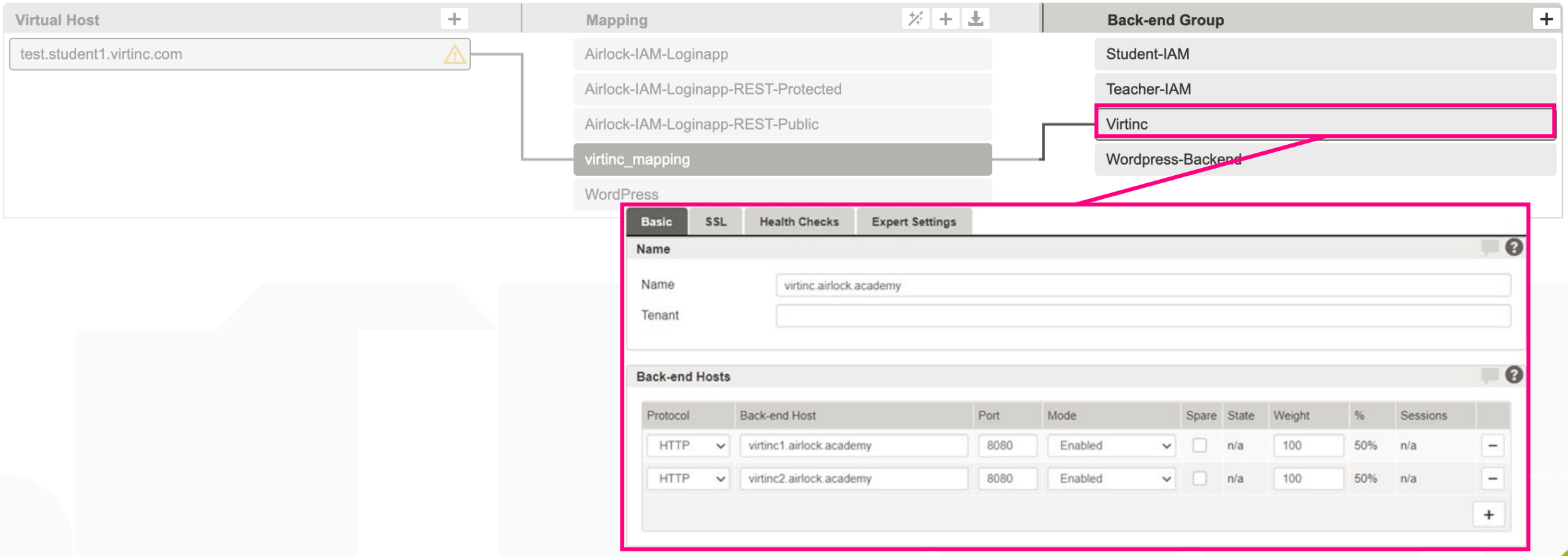
Load balanced Airlock Gateway farm – What to know

- Makes only sense, if „Quorum“ is 2. This means, that at least 3 Airlock Gateways are required for resilience!
- … or in a cloud deployment, because of "fixed" IP adresses.
- No config management optimization. Use REST-API instead.
- Hint: Use DHCP for front-end interfaces to apply VH configs.

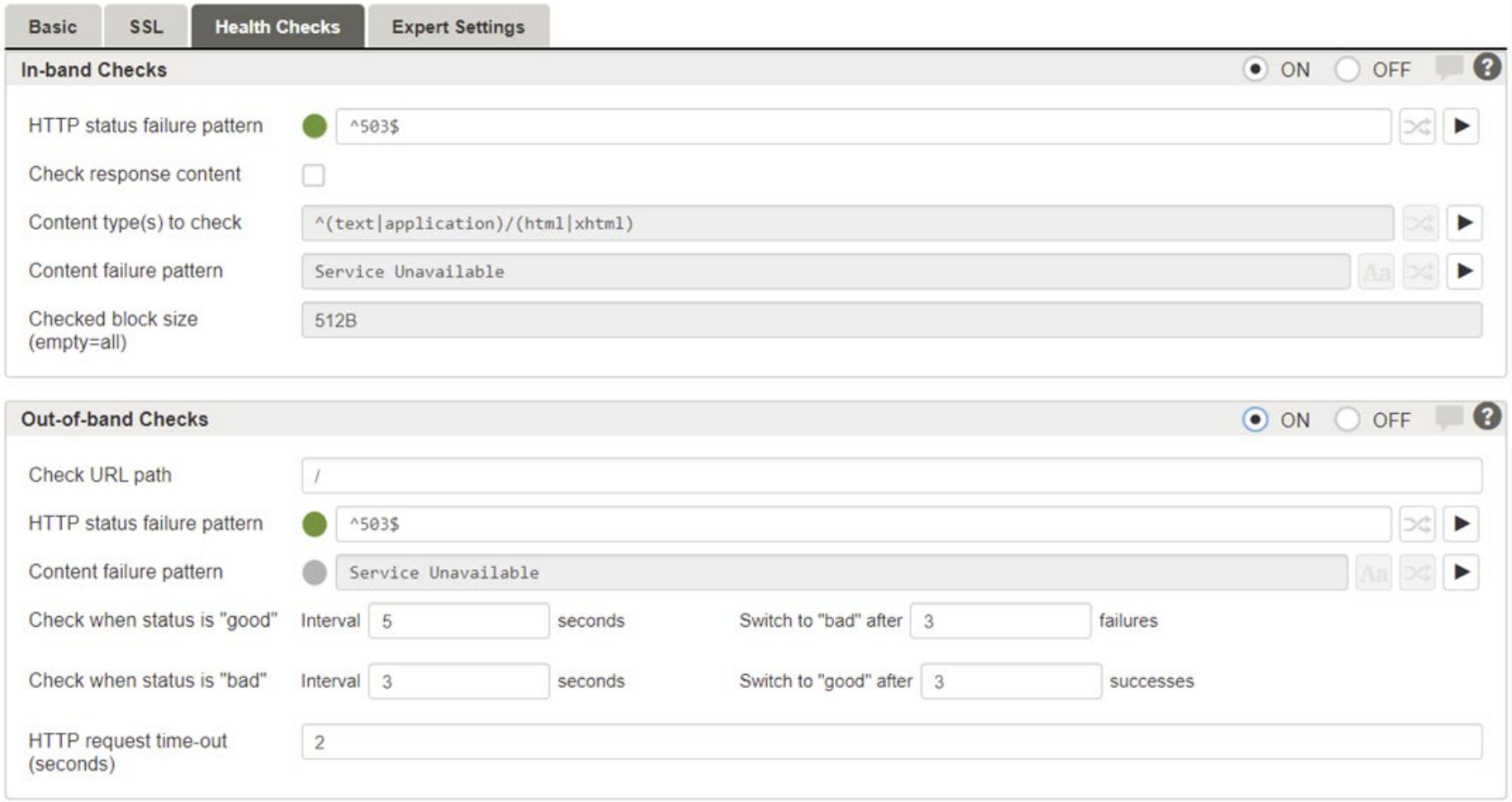
- https://github.com/airlock-presales/Airlock_Gateway_Sync

Reverse Proxy – Back-end Groups

A **collection of back-ends** which will provide responses



Reverse Proxy – Back-end Groups: Health Checks



Challenge: How to upgrade Airlock Gateways?

- https://docs.airlock.com/gateway/8.5/index/1648621057096.html
  «Failover cluster upgrade with full installation»

Challenge: How to upgrade back-ends?

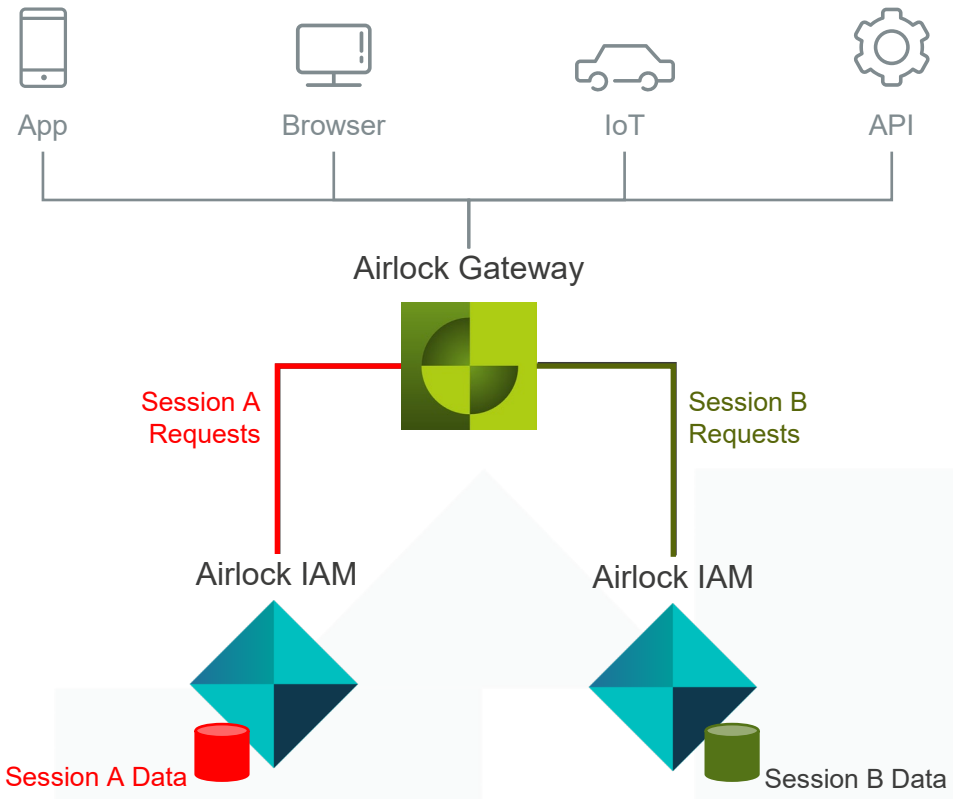
1. Set Back-end Host - Mode to «No more new sessions»
2. Activate
   → Reduces sticked sessions of this Back-end.
3. **Do your maintenance!**
4. Reset Back-end Host - Mode to «Enabled»
5. Activate

How to upgrade Airlock IAM back-ends?

- IAM is just another Back-end.
  - What's the only difference?

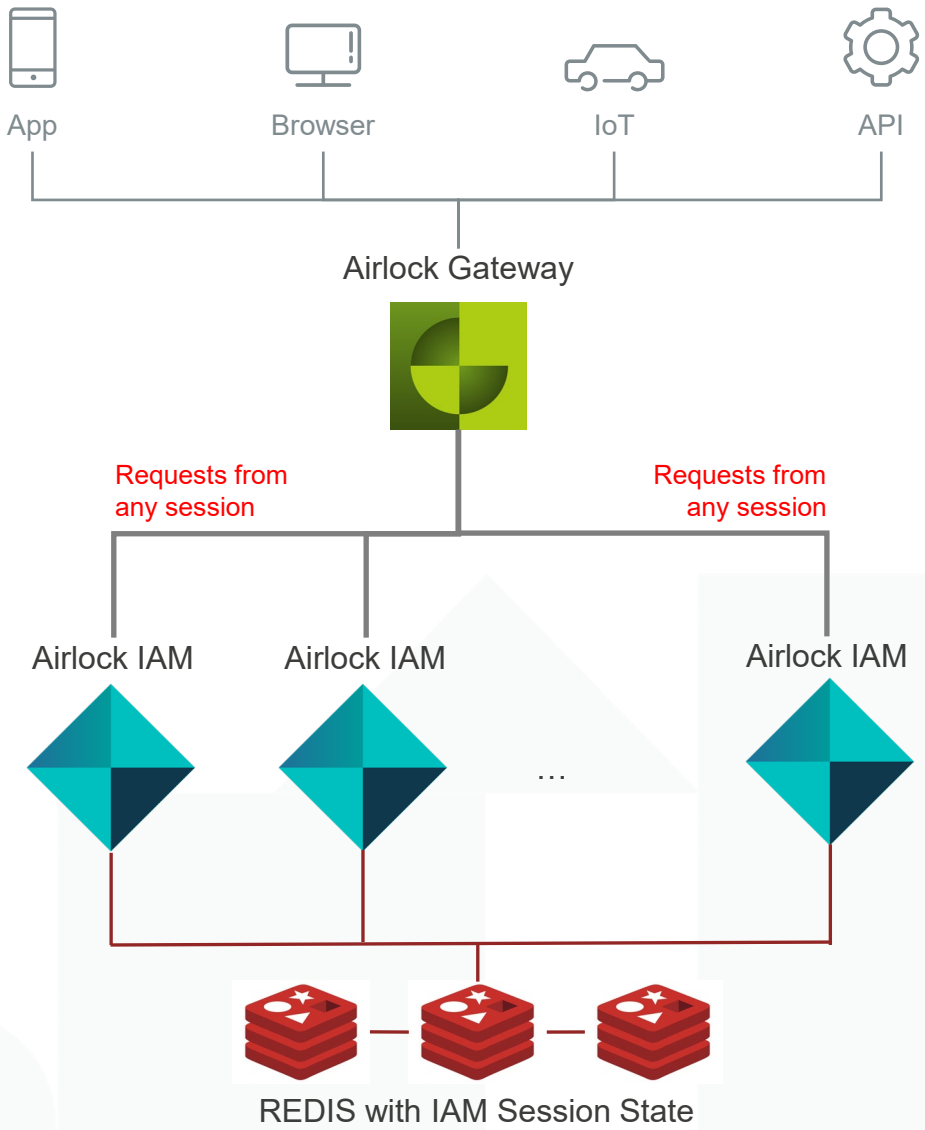
- Use Redis Sentinel!

Traditional IAM Setup



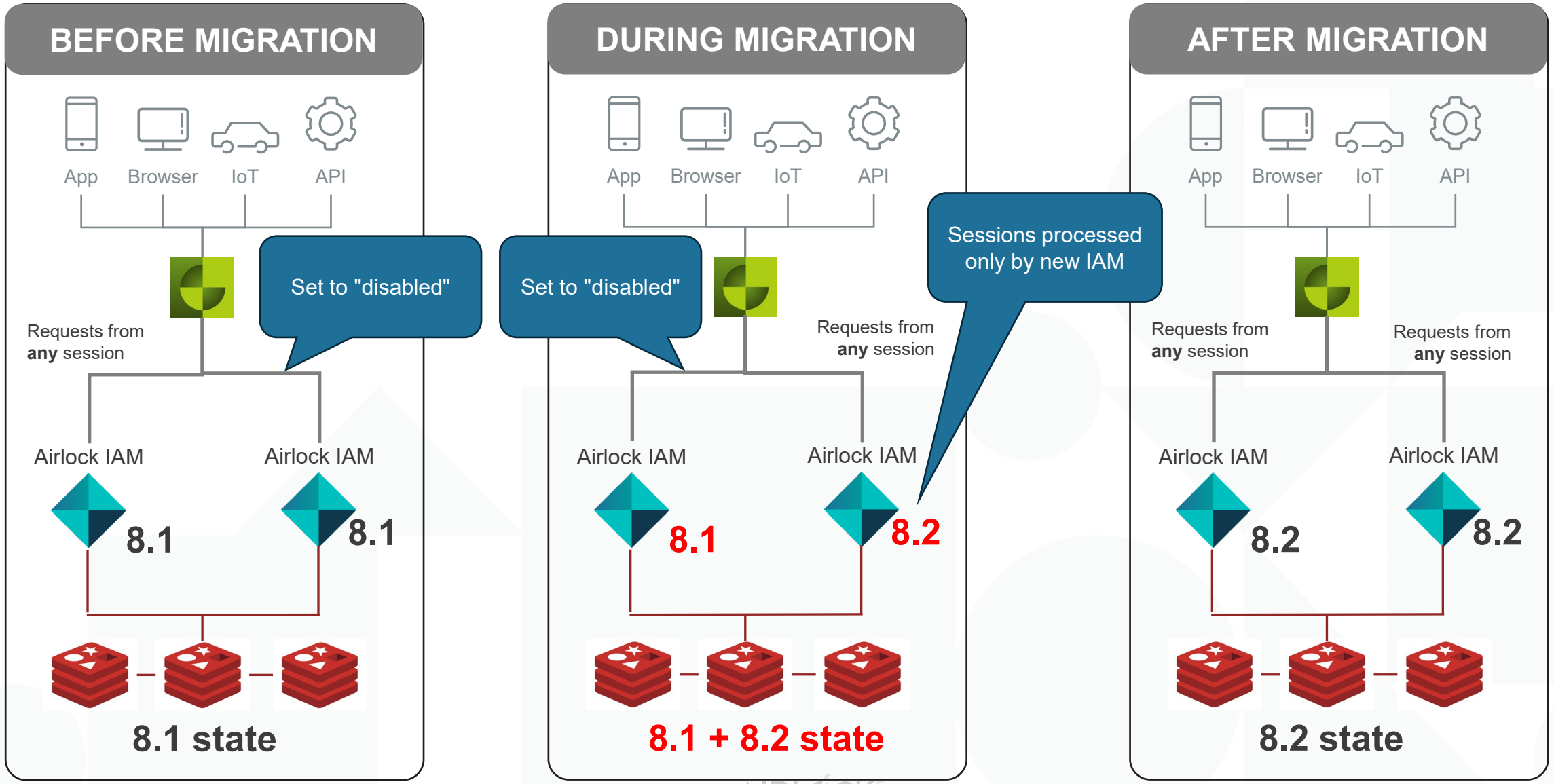
- Session stores are local to IAM instance
- Requires session-sticky load balancing
- No intra-session high availability
- May lead to session loss in case of IAM failure
- No dynamic horizontal scaling possible
- Still works with IAM >= 8.1

IAM 8.2 and newer: setup with externalized State



- IAM Session data can be stored in REDIS
- No more session-sticky load balancing
- Allows active-active setups
- Allows horizontal scaling

State Externalization: Updates at runtime



AIRLOCK®

Airlock

Questions?

Call us **+41 44 268 87 00**