

# Storms in the Cloud

Designing and using a fault injection system

Michalis Zervos  
@mzervos



## Xbox Live is down due to Microsoft Azure issues

JEFF GRUBB NOVEMBER 18, 2014 6:00 PM

TAGS: XBOX 360, XBOX LIVE, XBOX ONE

<http://venturebeat.com/>

## Salesforce outage continues in some parts of the US

CEO Marc Benioff took to Twitter to apologize to users  
<http://www.pcworld.com/>

Netflix CS   
@Netflixhelps

 Follow

We're aware some members are experiencing technical difficulties on the web site in all regions. We hope to resolve this asap.

3:54 PM - 3 Feb 2015

<https://twitter.com/netflixhelps>

## Gmail and Google Drive are down again for some people

[thenextweb.com](http://thenextweb.com) > Google ▼

Feb 1, 2016 - Google appears to be experiencing some issues at the moment as a number of users are reporting that both Gmail and Google Drive are down.

Google News - <http://thenextweb.com/>

## Mysterious Xbox Live outage leaves some of your games inaccessible

[mashable.com/2016/02/22/xbox-live-outage-february-2016/](http://mashable.com/2016/02/22/xbox-live-outage-february-2016/) ▼

Google News - <http://mashable.com/>

## AWS Sydney outage prompts architecture rethink

By Allie Coyne  
Jun 6 2016  
12:33PM

Customers consider multi-region redundancy.

<http://www.itnews.com.au>

## Azure cloud suffers multi-region outage

IT News

Azure storage, virtual machines, websites, Active Directory and the ... It confirmed the outages but did not provide detail on the cause.

Google News - <http://www.itnews.com.au>



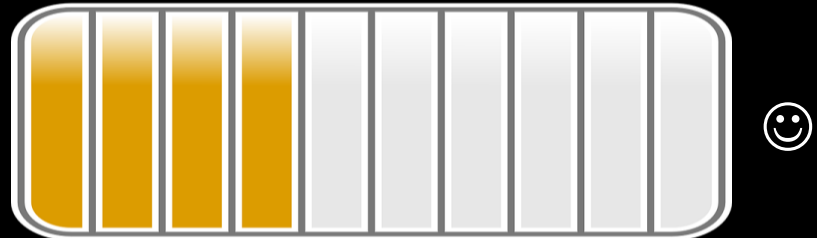
# Service Resilience

- Not a solved problem
- Goal is:
  - 100% uptime
  - No degradation
  - Responsive



# Traditional testing

- Unit tests
- Functional / Integration
- End to end



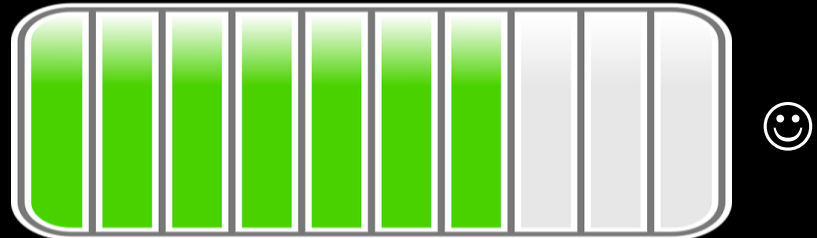
# Cloud services – Testing challenges

- Continuous evolution
- Multiple dependencies
- Global distribution
- Traffic fluctuation



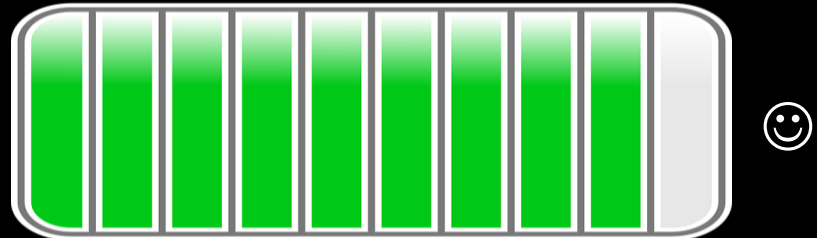
# Cloud services – Fundamentals

- Auto-scaling
- Redundancy
- Monitoring and detection systems
- Auto-mitigation / Failover mechanisms
- Staged deployments
- Data replication



# The extra mile

- Embrace failure
- Break the system
- Adjust the engineering process





Storms in the Cloud

# Fault Injection System

---

Support diverse services

---

Easy to use

---

Verify resilience and behavior

---

Simulate complex failures / real-life incidents

# Agenda

Designing a Fault  
Injection System



Usage patterns

# Faults

- Resource pressure
- Network
- Processes
- Virtual machine
- Platform
- Application specific
- Hardware

# Faults

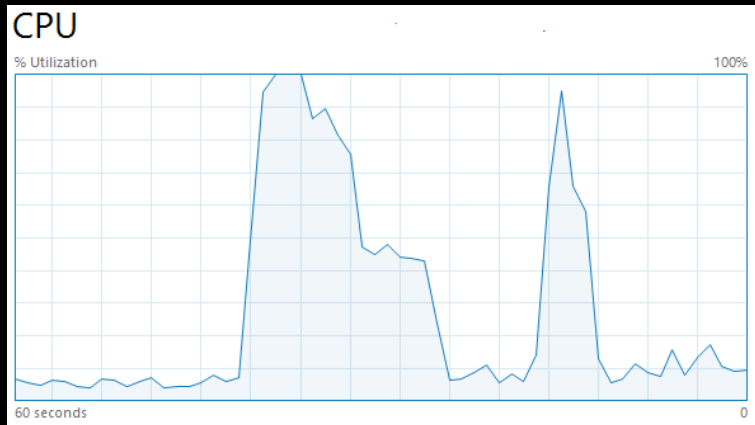
- Resource pressure
- Network
- Processes
- Virtual machine
- Platform
- Application specific
- Hardware

# Resource Pressure Faults

- CPU
- Memory
  - Physical
  - Virtual
- Hard disk
  - Capacity
  - Read
  - Write

## Available tools

- `consume.exe` (Windows SDK)
- `stress` (Unix)
- Sysinternals tools



# Faults

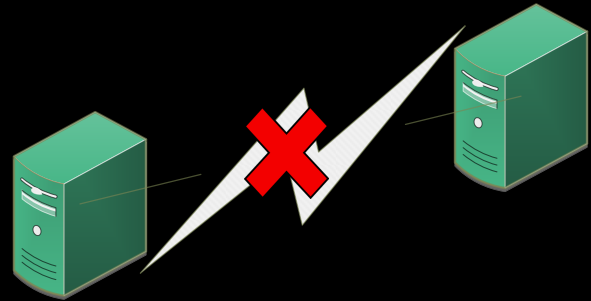
- Resource pressure
- Network
- Processes
- Virtual machine
- Platform
- Application specific
- Hardware

# Network faults

- Layers
  - Transport (TCP/UDP)
  - Application layer (HTTP)
- Types
  - Disconnect
  - Latency
  - Alter response codes (HTTP)
  - Packet reorder / loss (TCP/UDP)
- Filters
  - Domain / IP / Subnet
  - URL path
  - Port / Protocol

## Available tools

- Network Emulator Toolkit (NEWT)
- Fiddler core



# Faults

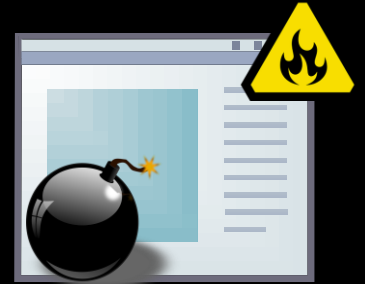
- Resource pressure
- Network
- Processes
- Virtual machine
- Platform
- Application specific
- Hardware

# Process faults

- Stop / Kill
- Restart
- Stop service
- Start
- Crash
- Hang

## Available tools

- OS commands
- Sysinternals tools



# Faults

- Resource pressure
- Network
- Processes
- Virtual machine
- Platform
- Application specific
- Hardware

# Virtual Machine / OS faults

- Stop
- Restart
- BSOD / Kernel panic
- Change date
- Re-image

## Available tools

- Cloud Management APIs
- OS commands
- Sysinternals tools

# Faults

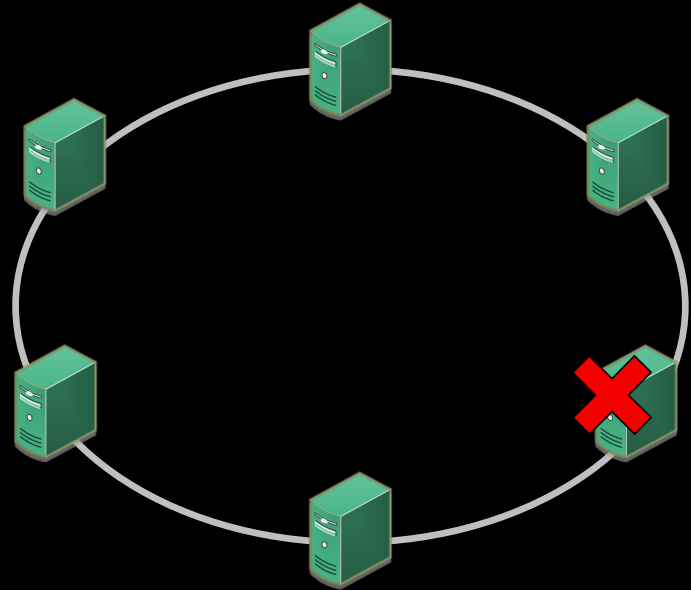
- Resource pressure
- Network
- Processes
- Virtual machine
- **Platform**
- Application specific
- Hardware

# Distributed platform faults

- Quorum loss
- Data loss
- Move primary node
- Remove replica

Available tools – Platform specific

- Service Fabric testability APIs



# Faults

- Resource pressure
- Network
- Processes
- Virtual machine
- Platform
- Application specific
- Hardware

# Application specific faults

- Hooks
  - Instrument service code
- Intercept / Re-route calls
  - No access to service code

## Available tools

- MSR Detours
- TestApi – Managed Fault Injection

```
public async Item GetItem(string id)
{
    var response = await this.client.GetAsync($"/items/{id}");
    if (response.IsSuccessStatusCode)
    {
        return await response.Content.ReadAsAsync<Item>();
    }

    return null;
}
```

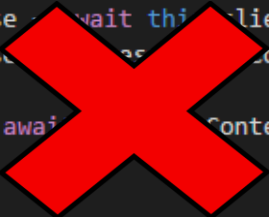
# Application specific faults

- Hooks
  - Instrument service code
- Intercept / Re-route calls
  - No access to service code

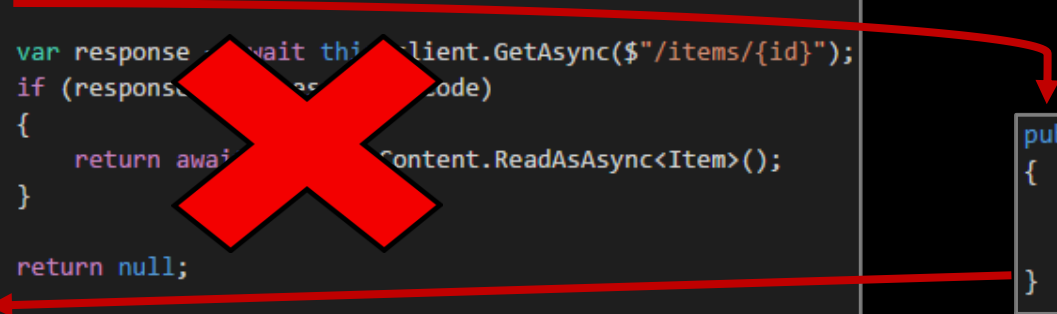
## Available tools

- MSR Detours
- TestApi – Managed Fault Injection

```
public async Item GetItem(string id)
{
    var response = await this.client.GetAsync($"/items/{id}");
    if (response.StatusCode == HttpStatusCode.NotFound)
    {
        return await Content.ReadAsAsync<Item>();
    }
    return null;
}
```



```
public void Intercept()
{
    Thread.Sleep(4000);
    throw new HttpRequestException("Not found");
}
```



# Faults

- Resource pressure
- Network
- Processes
- Virtual machine
- Platform
- Application specific
- Hardware

# Hardware faults

- Machine
- Network devices
- Rack
- UPS
- Datacenter

# Faults

- Resource pressure
- Network
- Processes
- Virtual machine
- Platform
- Application specific
- Hardware


# Injection mechanism

- VM External
- VM Internal – Service code external → Agent
- VM Internal – Service code internal → Hooks

# Injection mechanism

- VM External
- VM Internal – Service code external → Agent
- VM Internal – Service code internal → Hooks

# External injection



**Cloud  
Management Service**



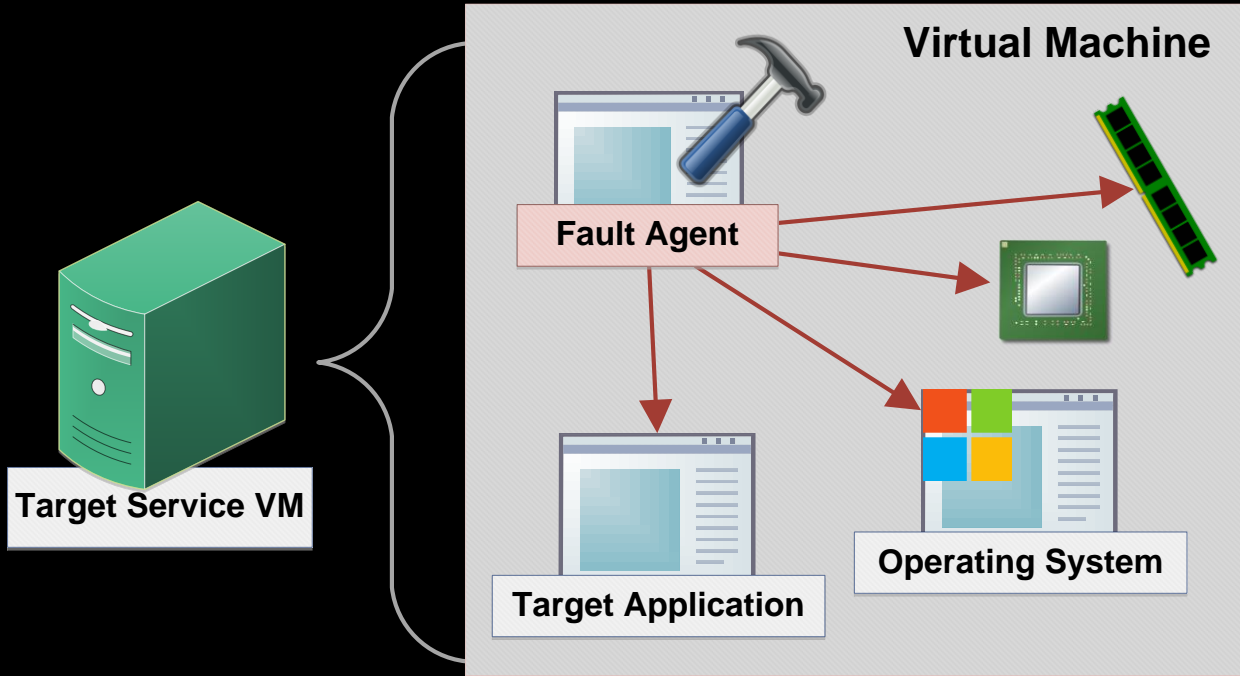
**Target VM**

- VM / Region Stop
- VM / Region Restart
- Re-image

# Injection mechanism

- VM External
- VM Internal – Service code external → Agent
- VM Internal – Service code internal → Hooks

# VM internal injection - Agent



- Resource pressure
- Network
- Processes
- OS
- Detours
- ...

# Injection mechanism

- VM External
- VM Internal – Service code external → Agent
- VM Internal – Service code internal → Hooks

# VM internal injection - Hooks



Target Application

```
public async Item GetItem(string id)
{
    var response = await this.client.GetAsync($"/items/{id}");
    if (response.IsSuccessStatusCode)
    {
        return await response.Content.ReadAsAsync<Item>();
    }

    return null;
}
```

- Application behavior
- Flexibility
- Service specific

# VM internal injection - Hooks

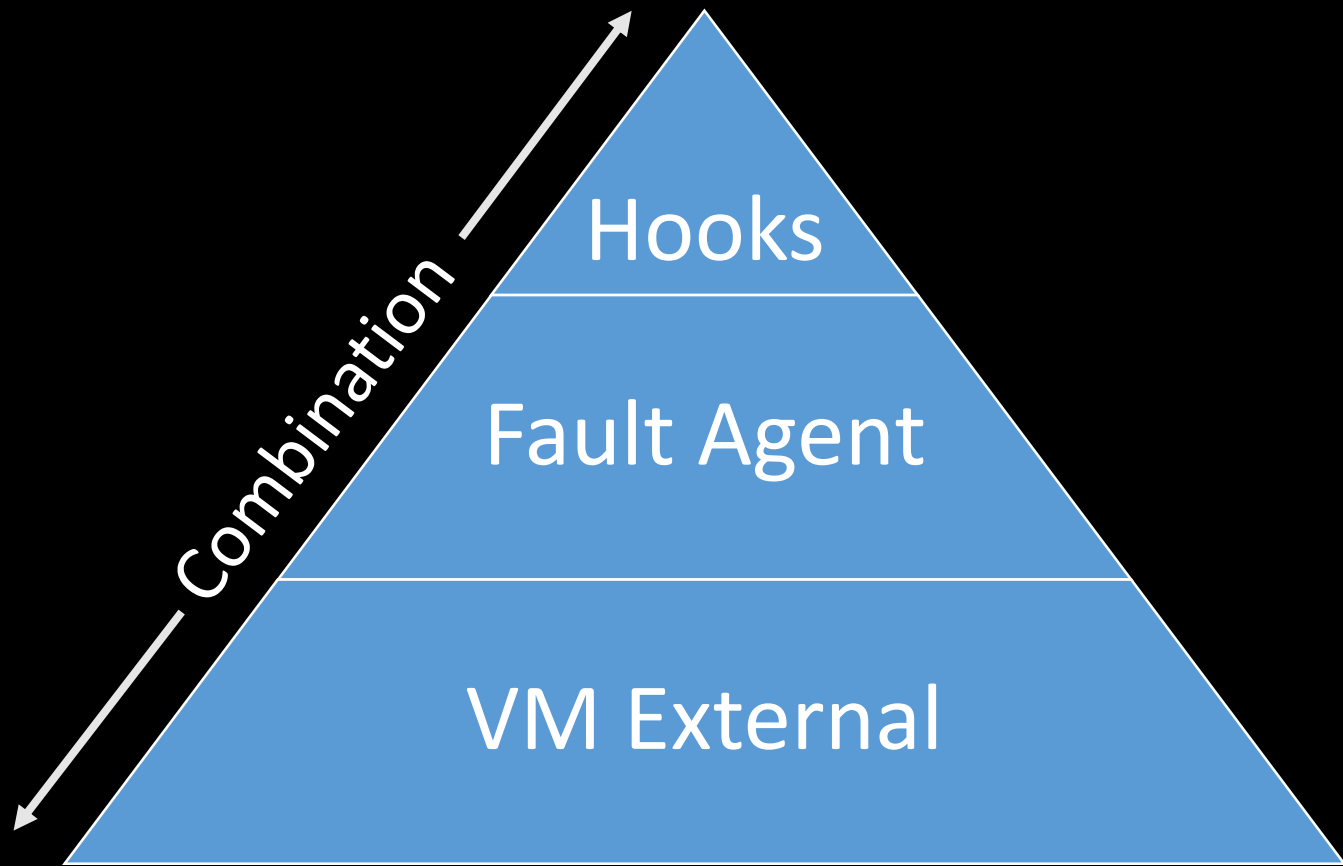


```
public async Item GetItem(string id)
{
    if (this.ShouldInject())
    {
        Thread.Sleep(4000);
        throw new HttpRequestException("Not found");
    }

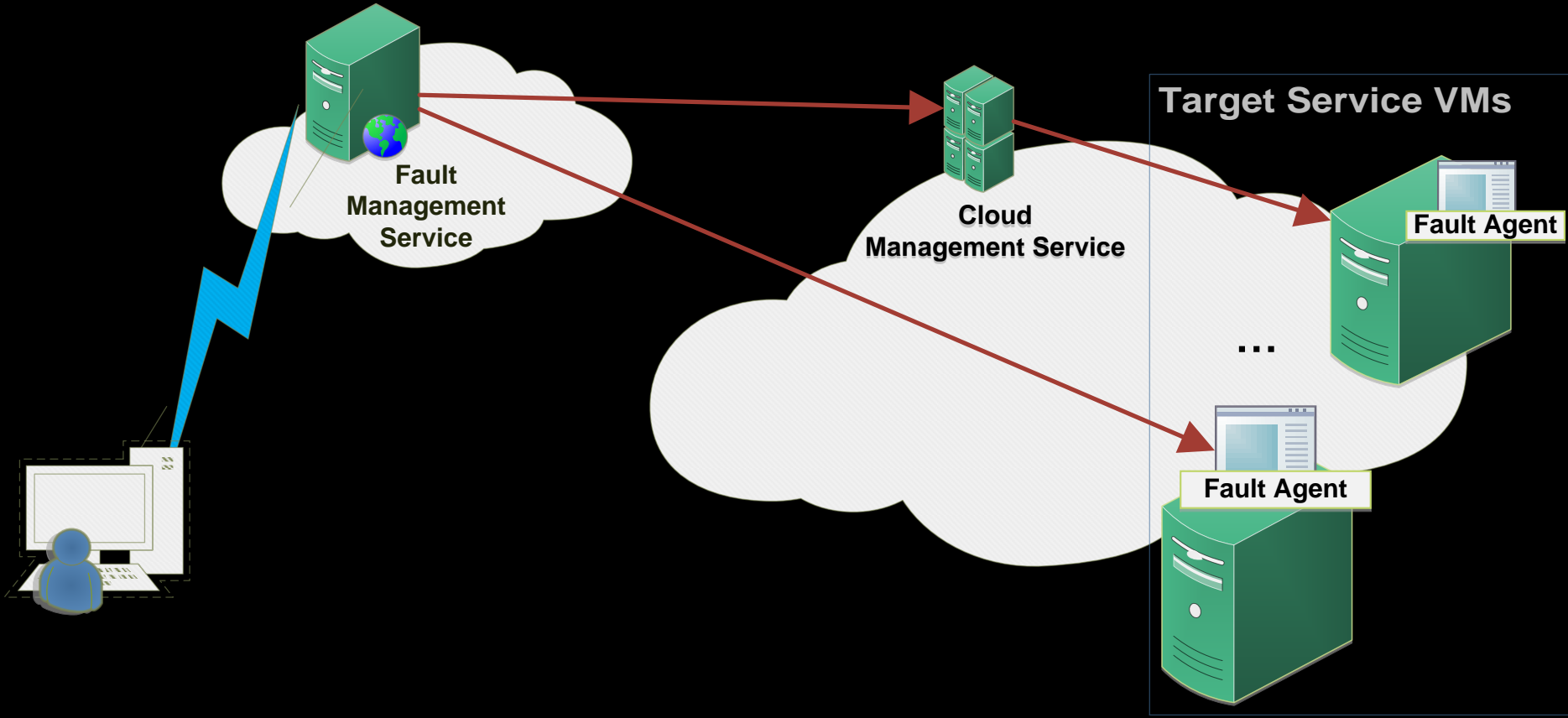
    var response = await this.client.GetAsync($"/items/{id}");
    if (response.IsSuccessStatusCode)
    {
        return await response.Content.ReadAsAsync<Item>();
    }

    return null;
}
```

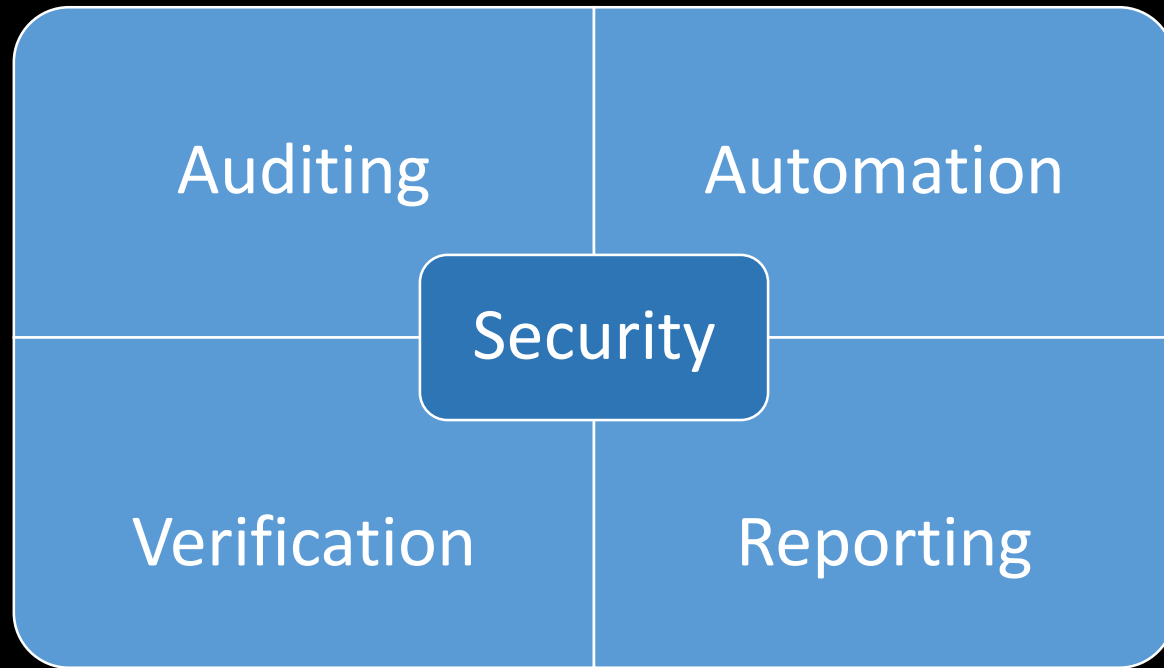
- Application behavior
- Flexibility
- Service specific



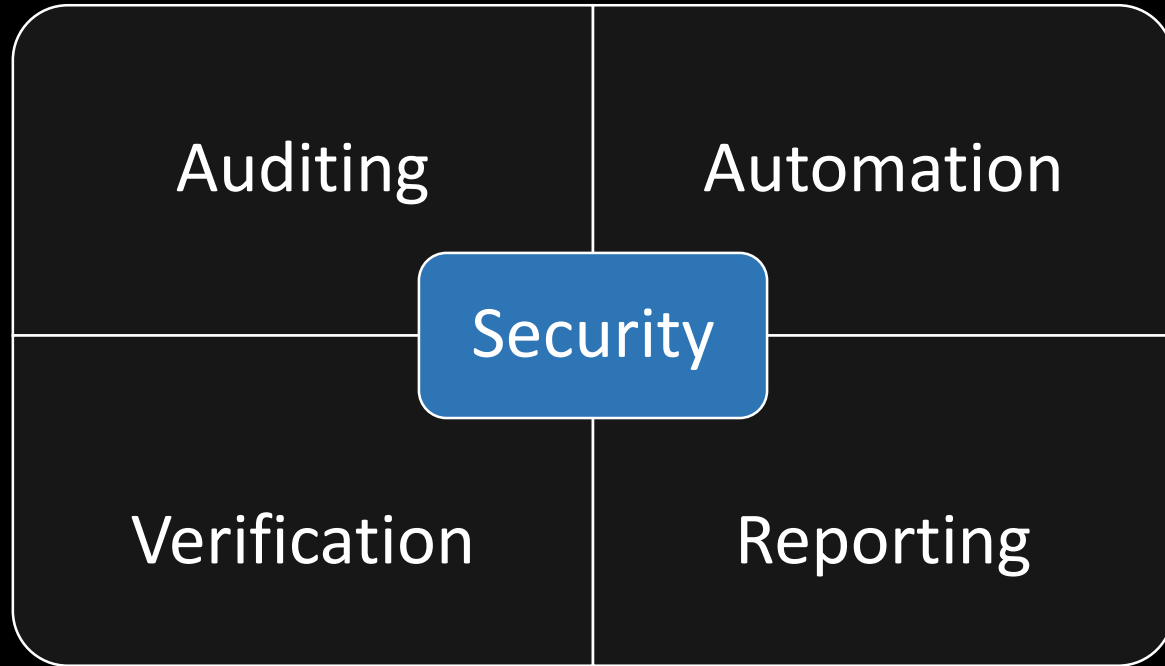
# System Architecture



# System components



# System components



# Security and Safety

- AuthN / AuthZ
- Fault agents
- Kill switch
- Safety nets

# Security and Safety

- AuthN / AuthZ
- Fault agents
- Kill switch
- Safety nets
- Integrate with Identity Provider  
Azure Active Directory
- Multi-Factor Authentication
- Least-privilege principle
- Granular access levels

# Security and Safety

- AuthN / AuthZ
- Fault agents
- Kill switch
- Safety nets
- Secure communication – TLS/SSL
- Code signing
- Execution permissions

# Security and Safety

- AuthN / AuthZ
- Fault agents
- Kill switch
- Safety nets



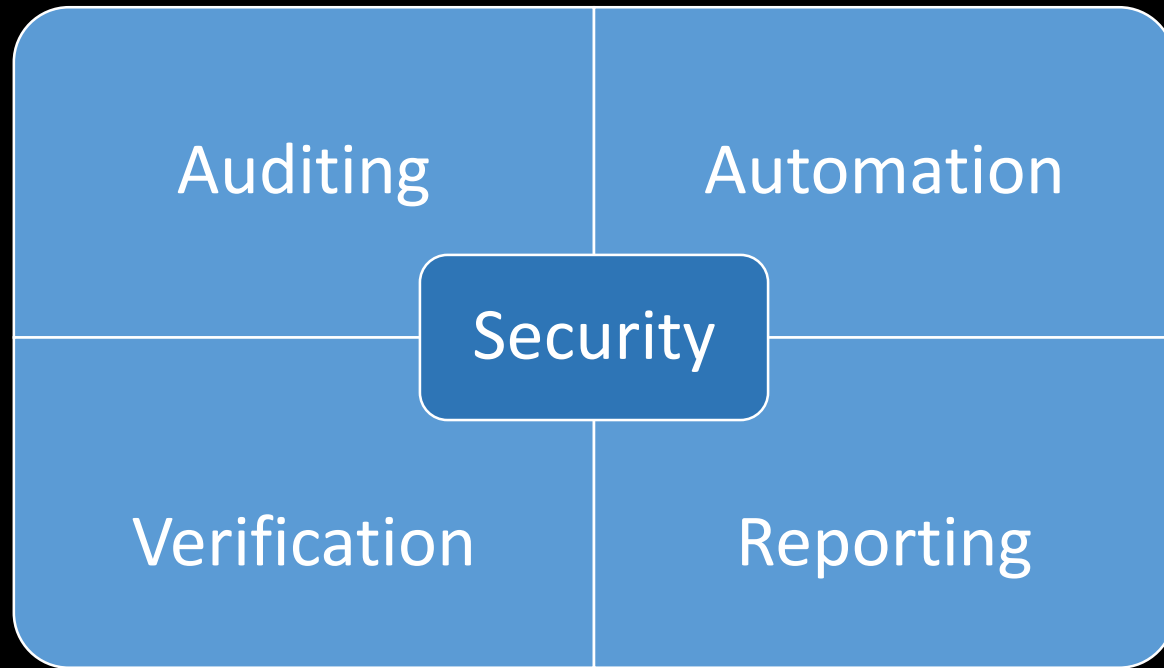
# Security and Safety

- AuthN / AuthZ
- Fault agents
- Kill switch
- Safety nets

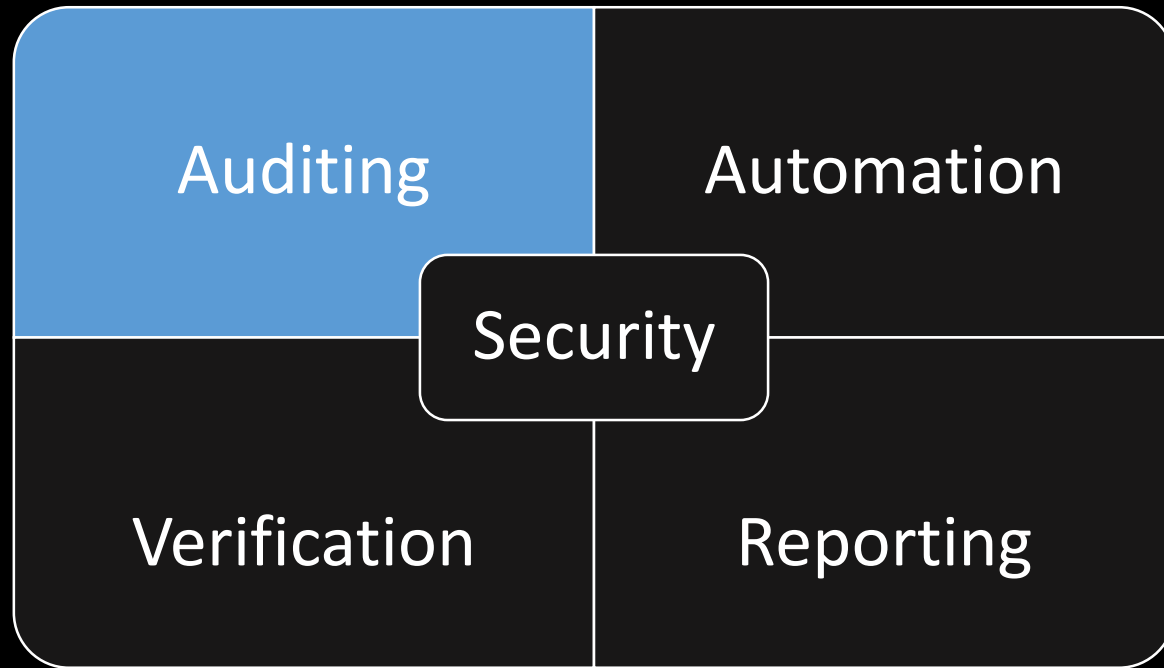
## **Auto fault removal**

- Agents – Service connectivity loss  
Agent-side detection
- Service malfunctioning  
Auto-monitoring module
- Unusual behavior  
Anomaly detection

# System components



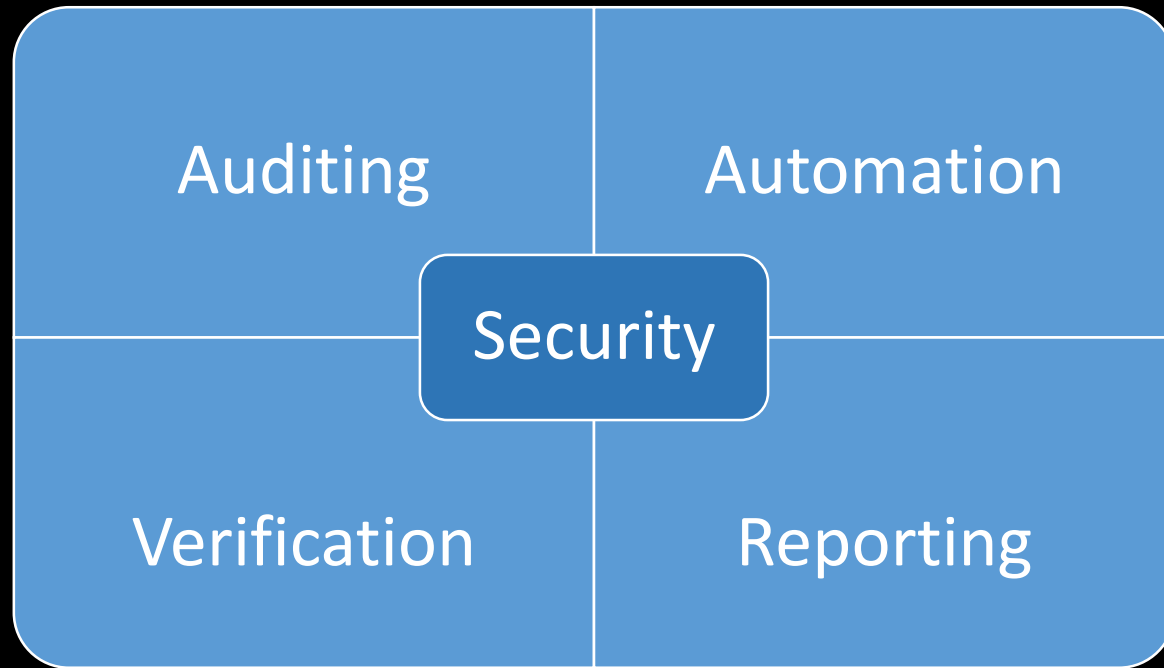
# System components



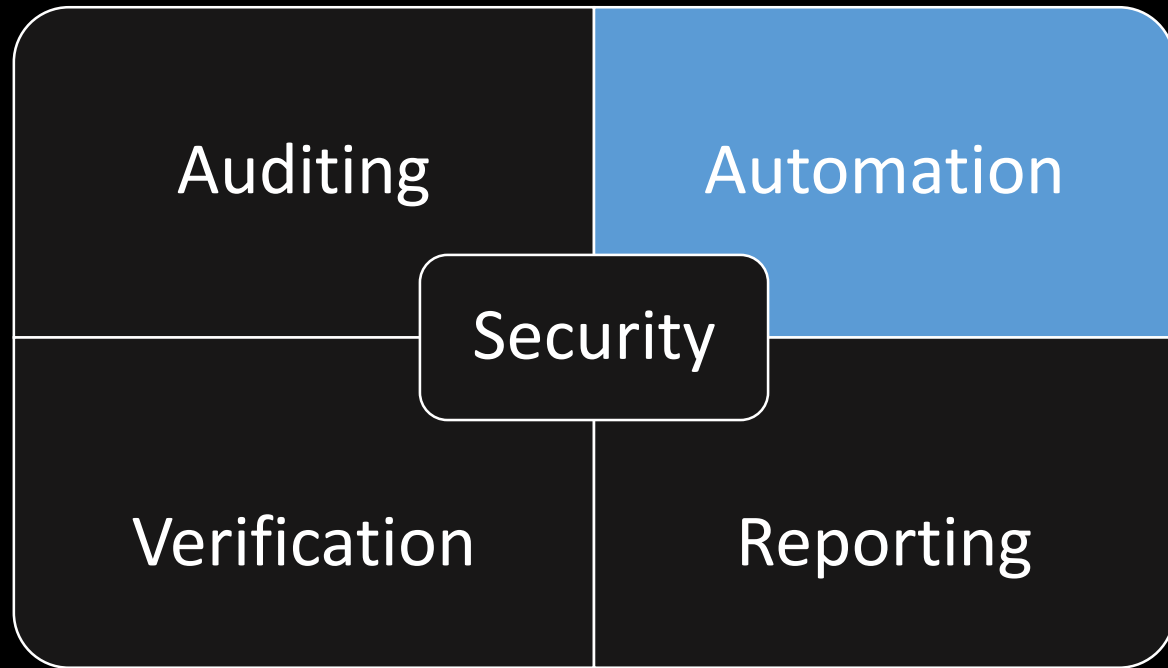
# Auditing

- Faults
- Fault agents
- Management service
- Clients

# System components



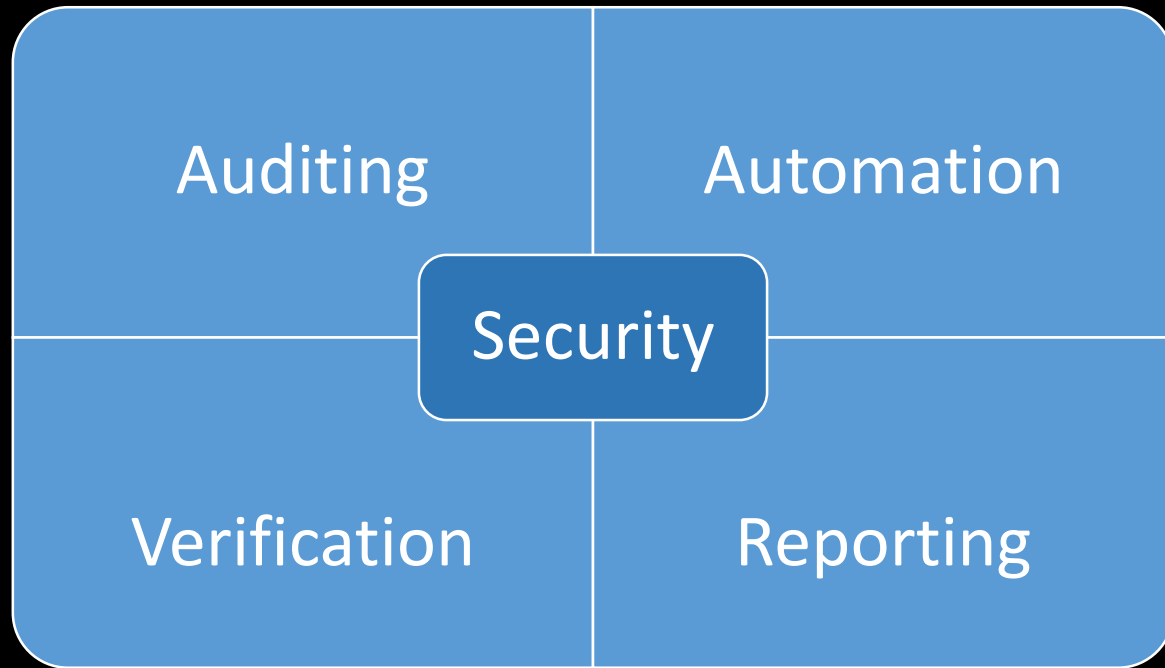
# System components



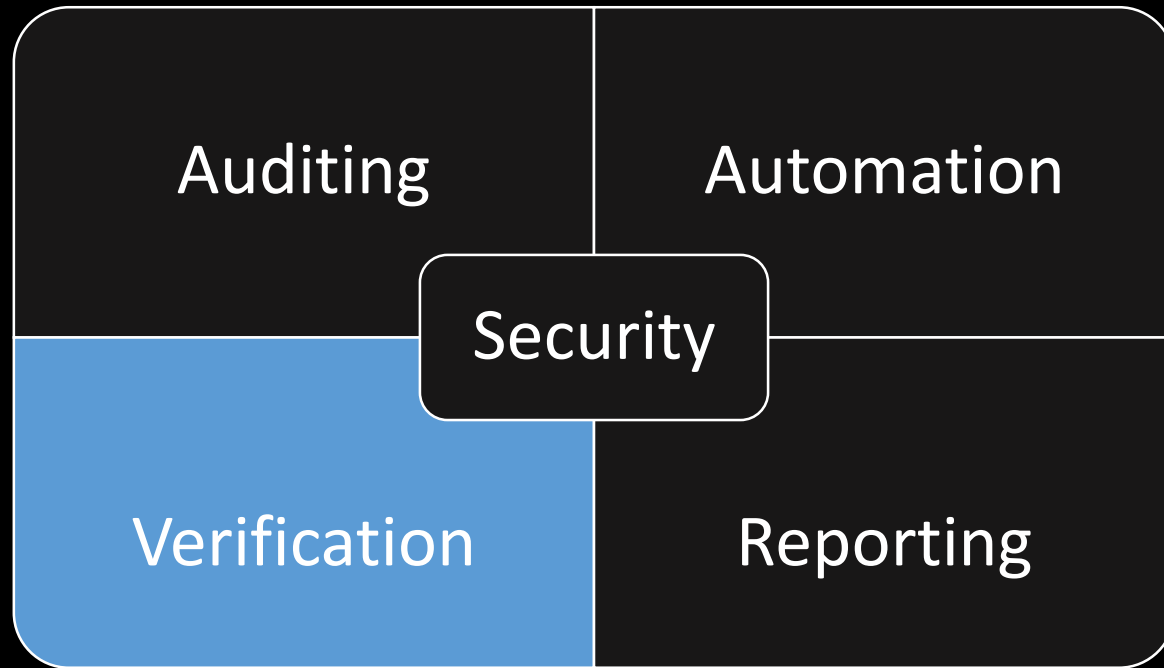
# Automation

- Scheduling
- Zero - configuration
- Dependencies auto-discovery

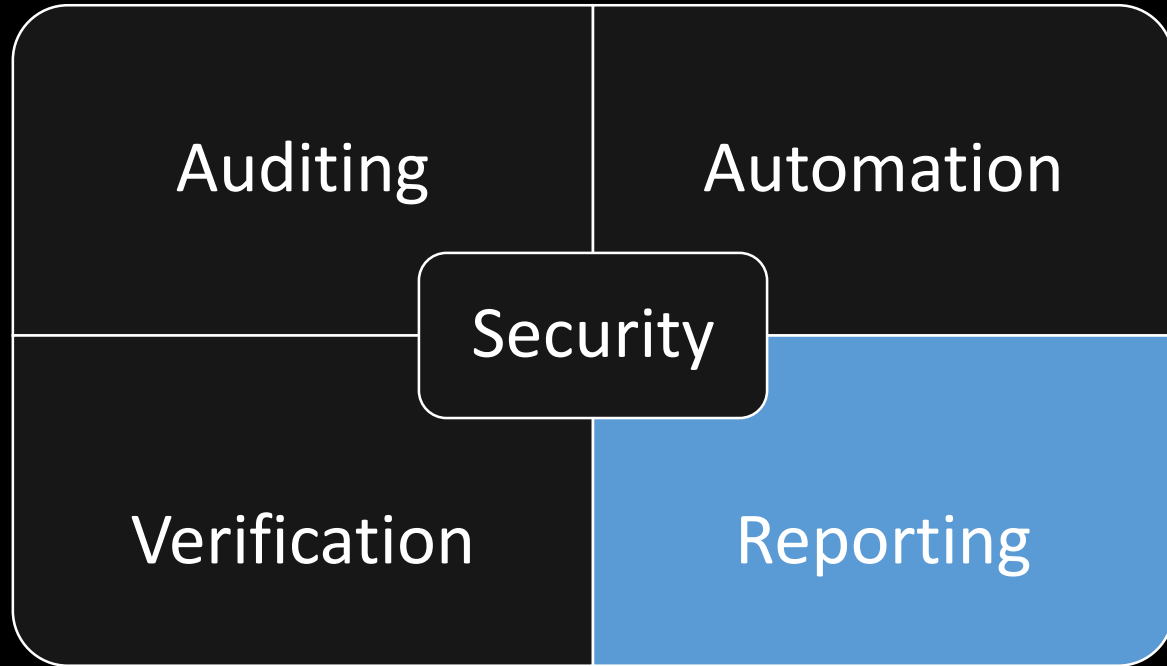
# System components



# System components



# System components





Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you. (0% complete)

If you'd like to know more, you can search online later for this error: HAL\_INITIALIZATION\_FAILED



# Usage scenarios

- Resilience verification
- Test new features
- Training
- Verify staged deployments
- Test detection, alerting, mitigation systems
- Repro incidents

# Injection environment



# Recovery Games



# Recovery Games

## **Attacker**

- Inject faults
- Provide hints

## **Defender**

- Assess
- Analyze
- Mitigate



# Recovery Games - Goals

- Familiarize with monitoring tools
- Recognize outage patterns
- Train on assessing the impact
- Root-cause / mitigation mindset
- Practice log analysis



# Invest in Fault Injection Testing



Resilience  
verification



Test new  
features



Training

Engineering process & culture