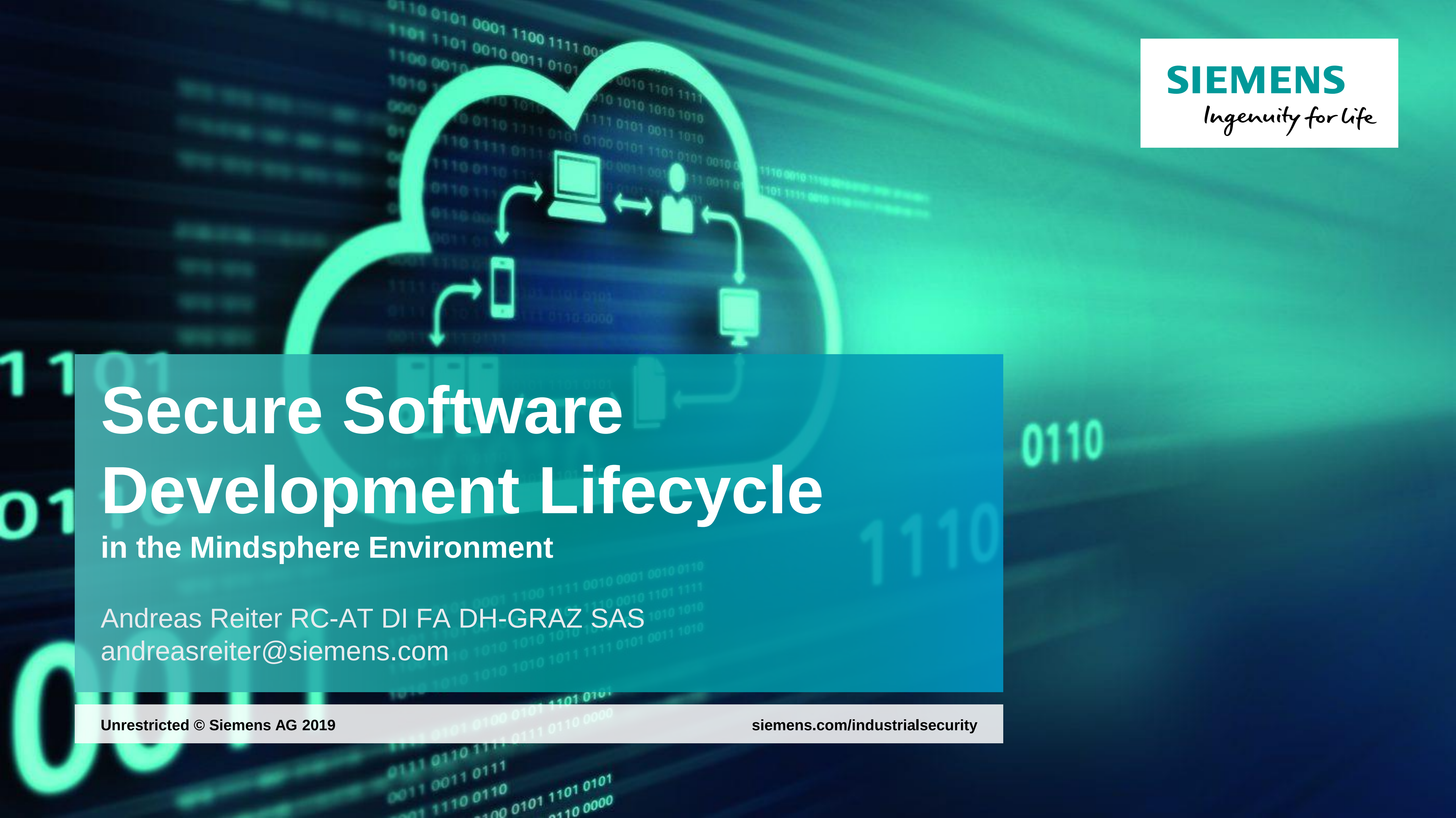


SIEMENS
Ingenuity for life

Secure Software Development Lifecycle

in the Mindsphere Environment

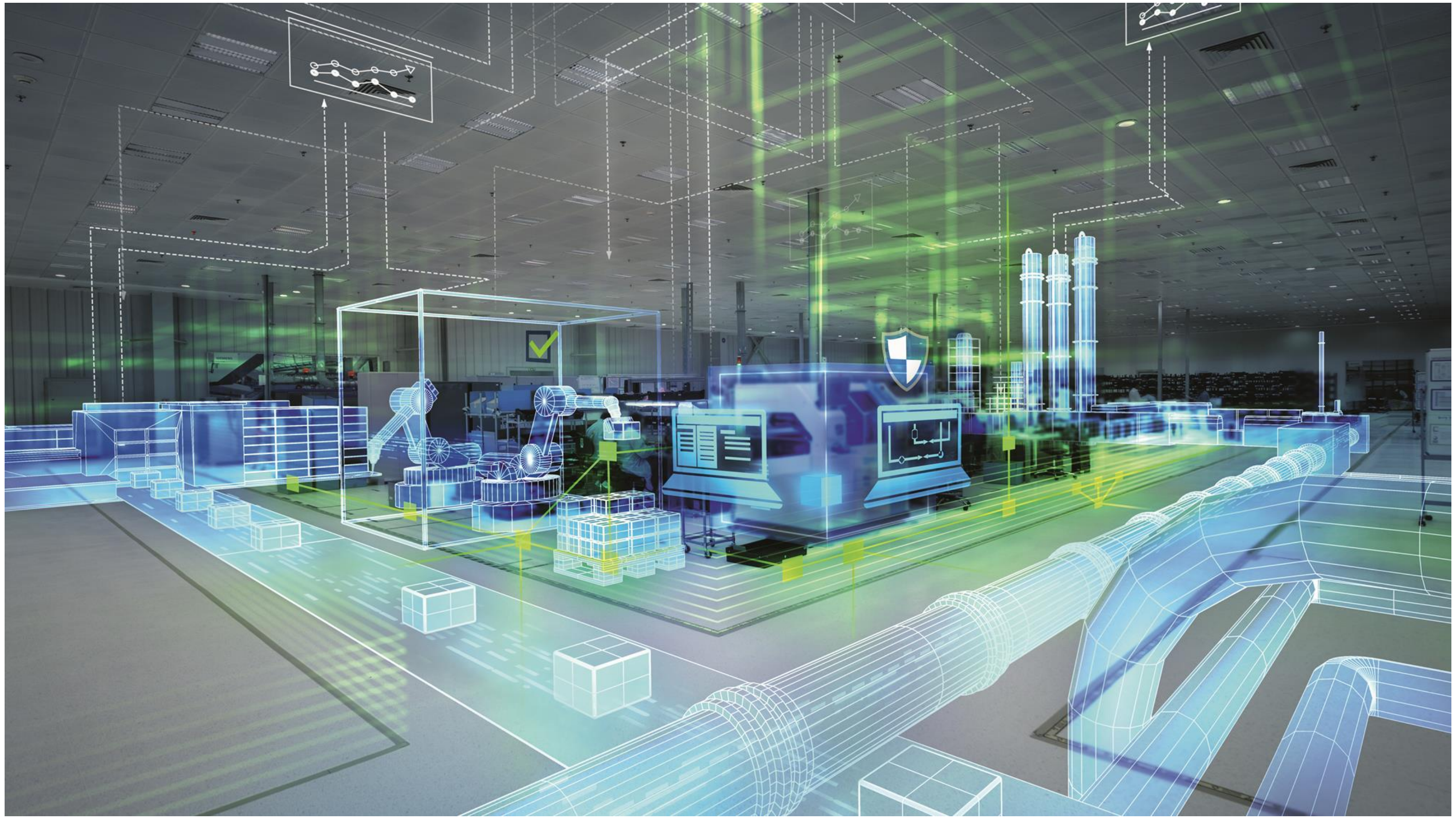
Andreas Reiter RC-AT DI FA DH-GRAZ SAS
andreasreiter@siemens.com

Factory Automation

SIEMENS
Ingenuity for life

- Massive amounts of data is generated

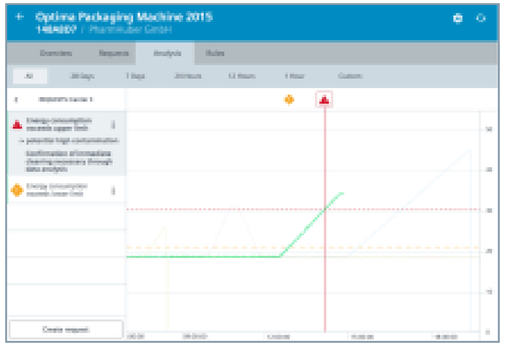
- Make use of this data



Mindsphere – IoT Operating System

SIEMENS
Ingenuity for life

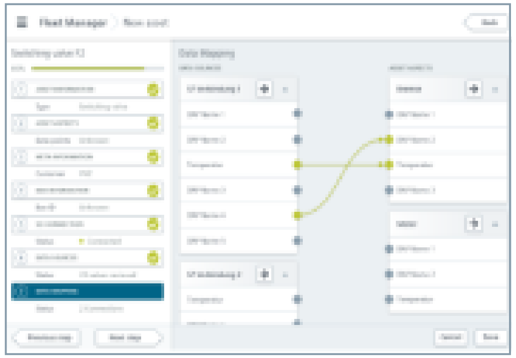
Information

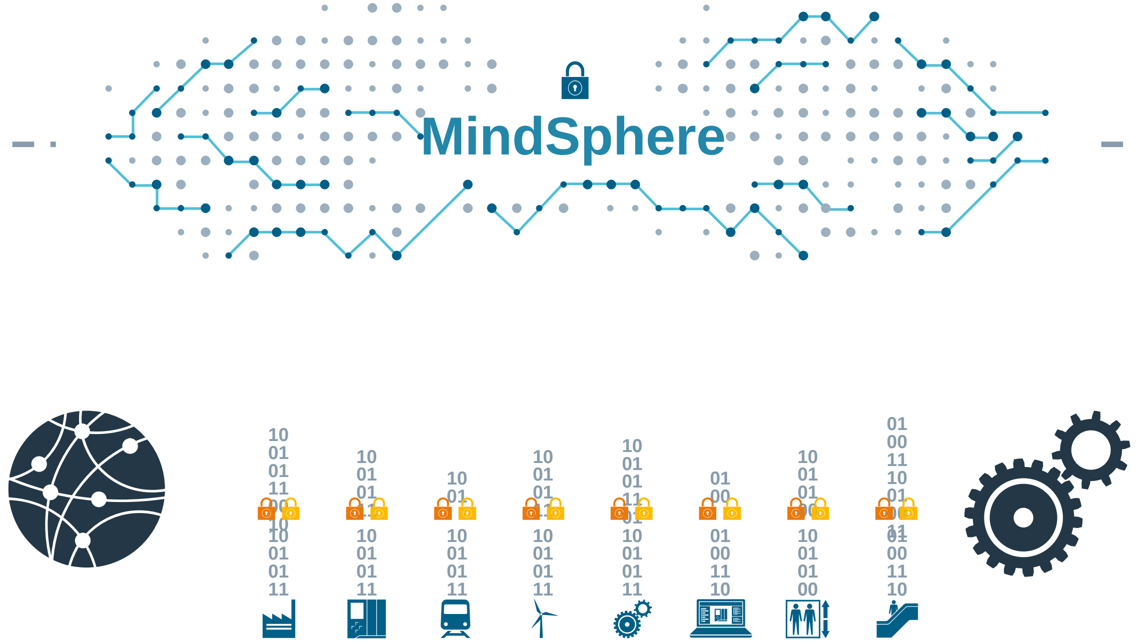


Virtual World

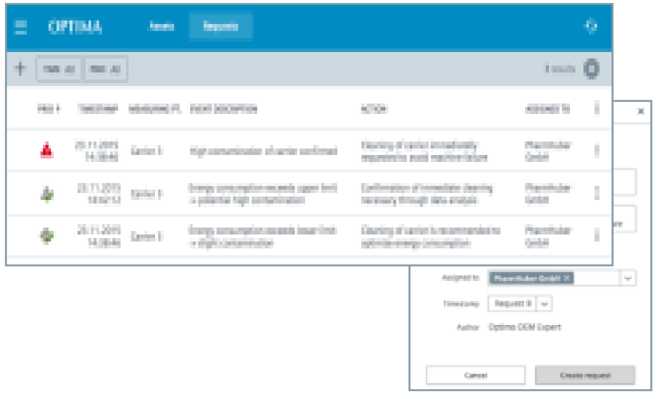
Real World

Data





Actions

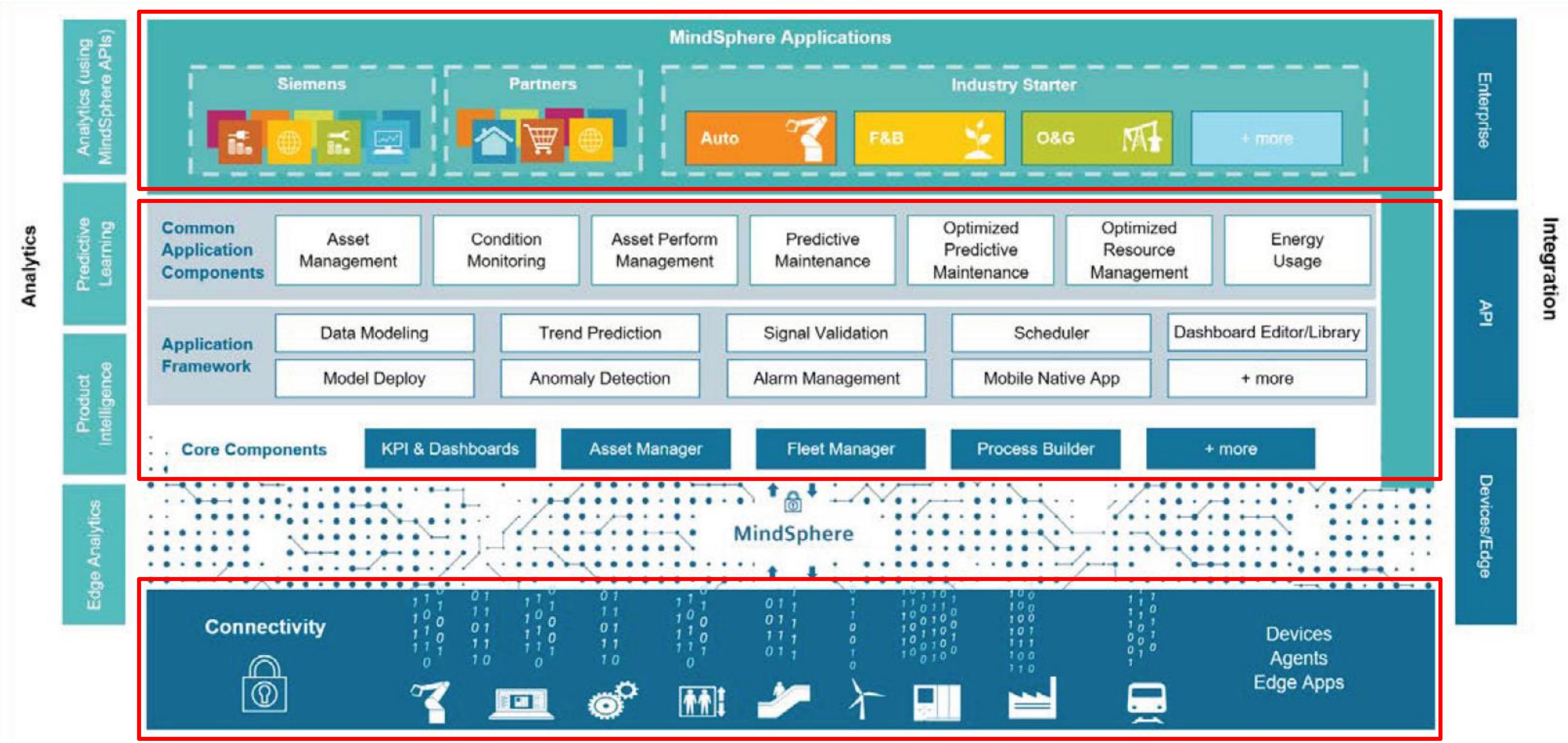


Risks

- Leaking company and production internals
- In case attackers can modify data, it can lead to wrong decision
- Safety implications

Mindsphere Architecture

SIEMENS
Ingenuity for life

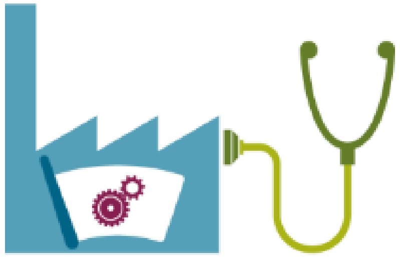


Mindsphere Applications

SIEMENS
Ingenuity for life

Real-time decision making

Predictive maintenance



Mindsphere Development

- Microservice architecture
- Development teams around the world
- Independent service development
  - Agree an APIs and interfaces
  - Versioning

Mindsphere Development

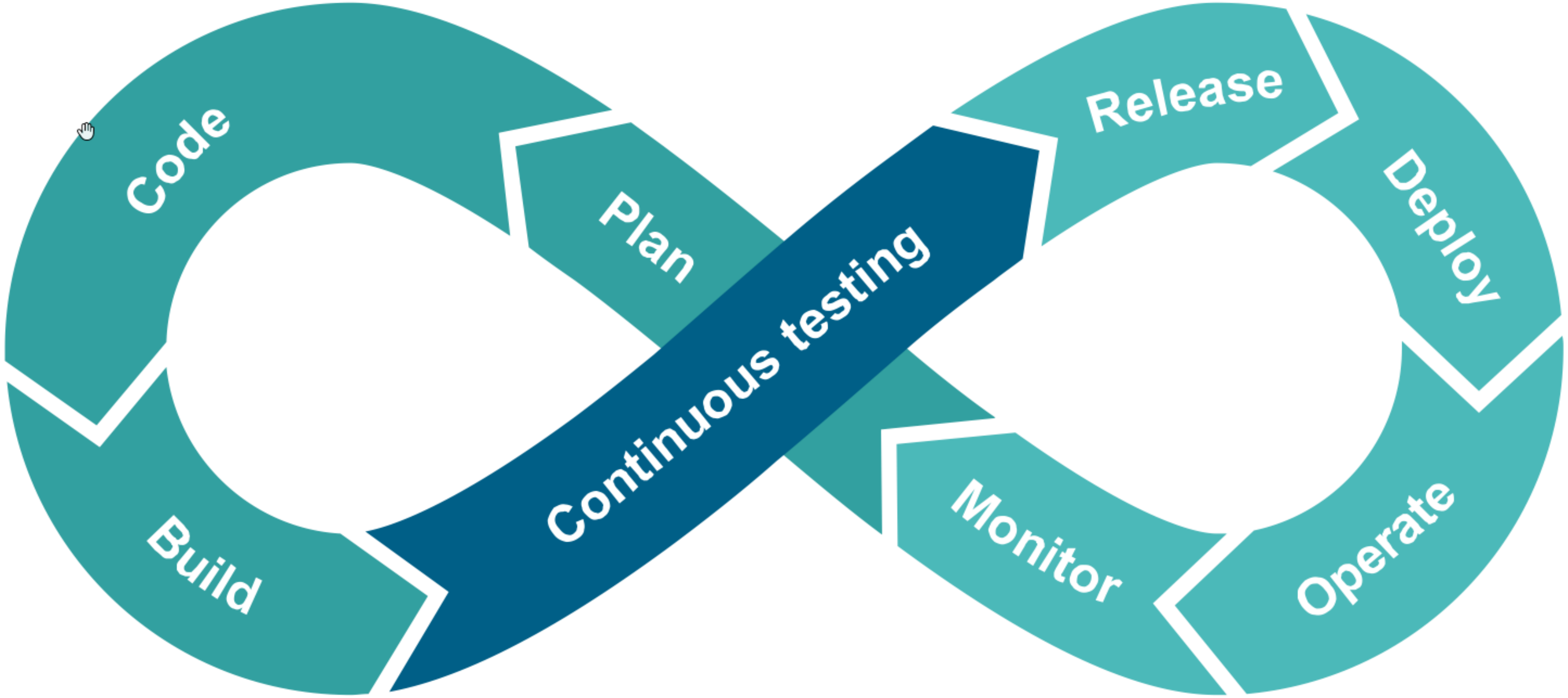
SIEMENS

Ingenuity for life

- **At the beginning:** Threat and Risk Analysis
- **Team starts engineering…**
- **Each team has individual automated pipelines for…**
  - **Testing**
  - **Continuous integration**
  - **Continuous delivery**

Process Overview

SIEMENS
Ingenuity for life



Secure Software Development Lifecycle

Costs / Effort	Costs / Effort	Costs / Effort
1	15	100
Development	Quality assurance/testing	Production

Process Overview

- **Continuous testing (and on regular schedules) involves…**
  - Behavior driven security testing
  - Test driven security
- **Security team implements security tests**
- **Development team implements security controls**

Behavior Driven Security Testing

SIEMENS
Ingenuity for life

- Behavior driven development approach for security testing
  - Bridge gap between behavior and implementation

```
Scenario: All incoming connections are TLS secured
  Given All open endpoints of target service are known
  When TLS configuration of all open ports is checked
  Then only secure TLS cipher suites are used
```

- Keywords control invocation: Given, When, Then, And,…

</> SAST

Static Application Security Testing

- Input parsing issues
- Injections
- Buffer overflows

sonarqube

Behavior Driven Security Testing

Dependency checking

- Find all sorts of issues in dependencies
- Checked on a regular basis

sonatype OSS Index

DEPENDENCY-CHECK

DAST

Dynamic Application Security Testing

- Runtime and environment related issues
- Authentication and authorization issues
- **Staging environments should be really close**

arachni
web application security scanner framework

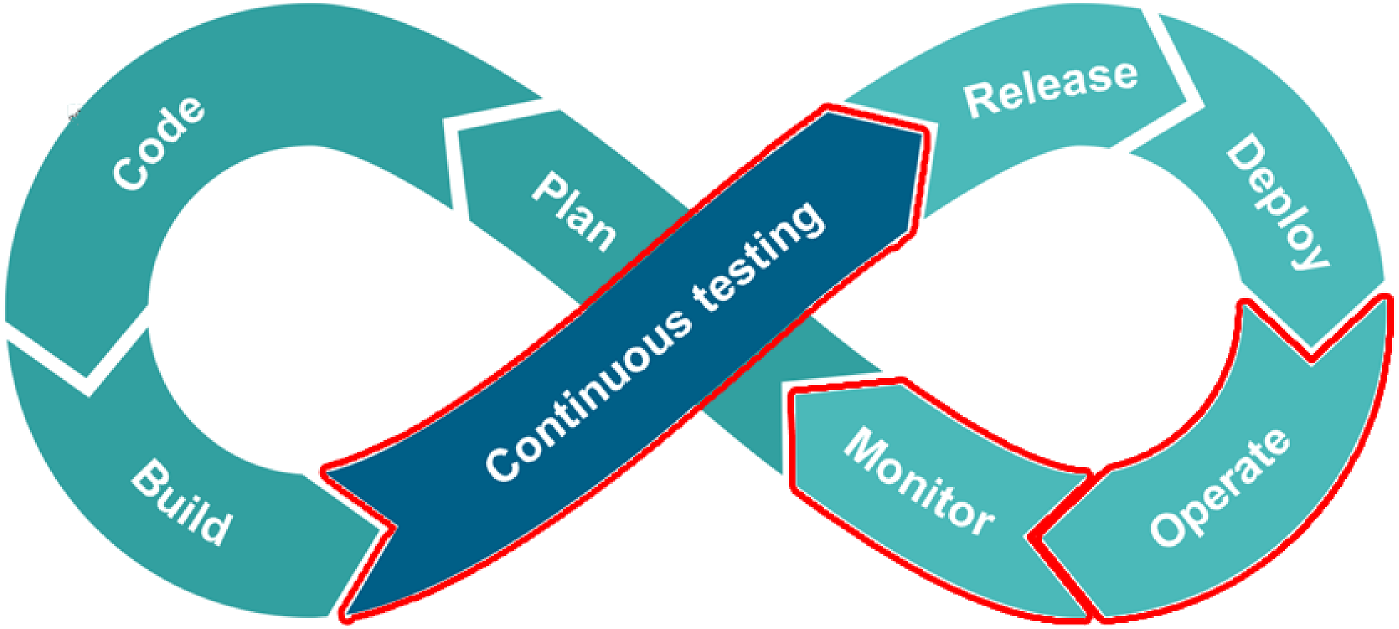
Behavior Driven Security Testing

Container Security

aqua

Infrastructure Security/Compliance

Back to the Process Overview ….



Unrestricted © Siemens AG 2019

DevOps to DevSecOps Success Factors

- **Automation**
  - Still provide efficient false-positives handling
- **Maintain the independence of the teams**
- **Provide immediate feedback**
- **Integrate in the teams’ development lifecycles.**

What’s Next?

- **Decouple automated security testing from CI/CD pipelines**
- **Security Testing-as-a-Service**
  - Local and hosted automated security testing on demand
  - Collect issues on a team’s dashboard

#1 DevSecOps: Don‘t skip design and architecture

Dev[Sec]Ops Pitfalls

#2 Security is part of the product

Dev[Sec]Ops Pitfalls

#3 Software lifecycle does not end after release

Dev[Sec]Ops Pitfalls

SIEMENS
Ingenuity for life

#4

Automation is not a replacement for audits or pen-testing

Contacts

SIEMENS
Ingenuity for life

Andreas Reiter

RC-AT DI FA DH-GRAZ SAS

E-mail:
andreasreiter@siemens.com

siemens.com/digital-factory