Software Architecture and Design Practices for Industrial IoT

Alisher Maksumov and Michelangelo Russo
GE Digital, General Electric

Saturn Conference, San Diego, CA
May 2016
2003 Northeast Blackout

- 50M people lost power for 2 days
- 11 people died
- $6B damages
Connected Devices

Need enormous gains in machine software development efficiency

Sources: Cisco, Harbor Research, http://www.futuristspeaker.com
What is Industrial IoT?

Expected annual revenues by 2020

![Image of Industrial IoT categories]

- **Wind Turbines**
- **Gas Compressors**
- **Gas Turbines**
- **MRI Machines**
- **Jet Engines**
- **Locomotives**
GE Gas Turbine Example

6 MAIN SENSORS TAKING IN 8,000 DATA POINTS PER SECOND
470 MW ENOUGH TO POWER 470k HOMES ANNUALLY

LOWER EMISSIONS FOOTPRINT

= 9,470 Cars OFF EUROPEAN ROADS*

981,000 POUNDS OF METAL = BOEING 747

630,000 HORSEPOWER = 1,650 F-350 TRUCKS OR 1,420 FERRARIS

* Under assumed operating profile of 4,000 hours per year, or an equivalent operating basis.
Critical Problems to Solve by IIoT

Key capabilities: asset connectivity, visibility, management, analytics, alerting
Architecture and Design Goals

Industrial Assets
Infinite streams of real-time data

Industrial Cloud
Infinite data storage and compute

Sensors → SCADA → Gateways → Routers → Networks

Responsive  Interoperable  Scalable  Easy to use
Fail-safe  Remotely manageable  Available  Secure

MM Data Steams → Command/Control
Architecture Approach

- **On the Cloud:**
  - Infrastructure – elastic, secure, available, VMs, containers
  - Microservices – separation of concerns, catalog, management
  - User Experience – domain specialized flows and patterns
  - User Interface – responsive, scalable, consistent
  - DevOps – development, testing, deployment automation
  - Security – infrastructure, apps/services, regulatory compliance
  - Legacy – support for existing legacy apps and services

- **On the Edge:**
  - Interoperability – industrial protocols
  - Security – holistic approach
  - Data collection – store/forward, transformation
  - Analytics – local processing
Hybrid Infrastructure in the Cloud

- Industrial Apps/SaaS
- API Gateway
- Microservices on CF
- Microservices on Docker
- Legacy Apps/Services
- Cloud Foundry
- Docker Containers
- VMs
- IaaS
- Cloud Hardware Infrastructure

- Hardware Infrastructure
- Run-time Services and Discovery
- Industrial Domain Services and Apps
Challenges and Learning

- **Architecture:**
  - SDK based legacy stack -> Cloud hosted microservices
  - Design patterns, APIs, standards, governance

- **Development:**
  - Scrum+Waterfall -> Pair programming, “pure” Scrum
  - OSGi+Java -> CF+Java, Go, Node.js, etc.
  - Best practices – 12factor app, configuration, performance

- **DevOps:**
  - CI/CD – testing, staging, deployment automation
  - Support – 24/7, online forum, phone, email, etc.
Challenges: Building The Edge Platform

The existing landscape:

1. **Hardware + Software Tightly Coupled** ➔ *Labor intensive update*
2. **Non Standard Interfaces** ➔ *Lack of interoperability*
3. **Lack of Scalable Architecture** ➔ *Limited analysis and processing of data*
4. **High number of Proprietary Software stacks** ➔ *Limited maintainability*
Lessons Learned…

• Platform must be:
  • As HW/OS-Agnostic as possible → Java, migrating to next-gen containers
  • Scalable → Footprint down to ~10MB
  • Pluggable → Service oriented architecture
  • Customizable → SDK

• Separation of concerns must be achieved between Real-Time (critical) components and non-Real-Time → Real Time Java ultimately dismissed
Functional Reference Architecture*

EDGE DEVICE

**Core Module**

- Application 2
- Application 3
- External Apps/Services

**External Apps/Services**

- Application 2
- Application 3

**APPLICATION**

CUSTOM, DOMAIN SPECIFIC

**Cloud Connectivity**

**External App Communication**

**Human Connectivity**

**Device Management**

**Data Management**

**Machine Connectivity**

**Core Services:** Logging, Permission Control, Troubleshooting

**Core Module**

**Virtualization Layer (Safe & Cooperative Communication)**

**Non-RTOS**

**RTOS**

**REAL-TIME CONTROL LOGIC**

* = Example of a possible deployment
Challenges: Building The Security Infrastructure

- Different security paradigms from edge to cloud → Isolated vs. Shared services
- Control on full stack (HW+SW) is not always an option
- Aging, unsecure protocols still used in the field
Lessons Learned...

• Security is an integrated story: designs, processes and practices must coordinate → *Device-initiated communication*

• Certificate-based infrastructure should be preferred to user-based authentication → *Availability of a signing authority*

• Airtight isolation is an illusion... → *Advocating federated solutions*
Edge Security Integrated Architecture

- Secure Protocols
- Perimeter Defense
- Trusted Computing
- User & Device Authentication
Risks due to operational needs leave attack surfaces open to exploitation
Compromise of any network participant threatens the system
Network itself is vulnerable

Leverages Virtualization approach
Risks due to operational needs can be accommodated minimizing attack surface
Compromise of any network participant is much harder
Compromised devices less of a threat to the system
Network is more secure
Takeaways

• Building a solution for Industrial IoT requires a platform that can stretch from sensors and embedded devices to elastic cloud infrastructure

• Use of microservices architecture and design patterns, 12factor app principles, security patterns, and devops automation are fundamental to our success

• Security has to be designed into hardware and software using holistic approach