Bosch Software Innovations
Spearheading the Internet of Things

Market presence
8.2m
connected devices using Bosch IoT Suite

Know-how
600+
IoT experts around the world
(Germany, Bulgaria, Singapore, China, Japan, USA)

Experience
250+
IoT international projects in the areas of manufacturing, mobility, energy, home & building, city, agriculture …

Mobility
Solutions for electromobility, intermodal transportation, and connected vehicles

Industry
Solutions for connected manufacturing

Agriculture
Solutions to support the sustainable intensification of food production

Energy
Solutions for smart and simple energy management

Smart Home & Building
Solutions for connected homes and commercial buildings

Smart City
Connected solutions for urbanites to make life easy and efficient
The Bosch IoT Suite
Technology for the connected world

**8.2 million**
Connected products

**5 million**
Connected cars
+0.2 million per month

**1 million**
Smart Homes
158 supported device types
only in one customer project

**26**
Number of supported protocols

**170**
Intelligent cloud
Bosch projects currently use the Bosch IoT Cloud

**5**
Multi-cloud strategy
Cloud providers x regions already supported. 2 more in the pipeline.
Autonomous Driving
Bosch leads the Patent Race

Who Leads the Autonomous Driving Patent Race?
Number of worldwide patent filings related to autonomous driving (January 2010–July 2017)

- Bosch: 958
- Audi: 516
- Continental: 436
- Ford: 402
- GM: 380
- BMW: 370
- Toyota: 362
- Volkswagen: 343
- Daimler: 339
- Google: 338

Based on a total of 5,839 patent filings related to autonomous driving identified and analysed by the Cologne Institute for Economic Research. Sources: Cologne Institute for Economic Research, WIPO.
New Bosch 300 Millimeter Semiconductor Fab in Dresden
Chipsets for IoT and Autonomous Driving
Bosch and Daimler Joint Pilot Project in San Jose
Automated on-demand ride-hailing service
Bosch Autonomous Shuttle
Driverless, electric, and seamlessly connected
Autonomous Driving, C2X, EVs…
Software Complexity is growing dramatically
Over the Air (OTA) Updates

OTA capability becomes an essential prerequisite for the success of the software defined vehicle!
- Must be able to support the dynamic evolution of complex, distributed IT systems
- System architectures must combine OTA capabilities with updatable hardware like FPGAs
- Must be able to deal with heterogeneous hardware and software environments
- Complex variations and variants
- Large scale: millions of connected vehicles
- Global availability must be ensured

Software + Content
- Bug Fixes
- New Features
- Forced security updates
Over-the-Air-SIG Goals

• Create **enduser-validated** requirements for the development of holistic OTA solutions for **various IoT domains**, starting initially with **automotive**
• Initiate technical validation projects for these requirements, e.g. through testbeds or challenges
• Initiate new industry standards to help harmonizing the OTA landscape
• Provide an efficient platform for OEMs, suppliers and industry organization to jointly shape the future of OTA solutions
Why OTA-SIG starts with the automotive industry?

• Based on Initial discussion with a German Car OEM
  – OEMs already do OTA to some extend
  – Often limited to update maps and certain ECUs

• Todays’ Major Challenges
  – With connected & autonomous cars the amount of software is dramatically increased
    • Not to mention business and entertainment services
  – Feature updates in cars not yet possible (except for Tesla)
  – Legacy systems in use prevent car manufacturers to properly rollout OTA
  – Proprietary
  – Knowledge about software artefacts are in authoring tools
  – No dependency management at deployment time
Efficient management of dependencies between ECUs is a key success factor for OTA

Especially once the plethora of low-end, highly specialized ECUs will be integrated into the OTA platform

Each ECU must provide dependency information via its own manifest
Admin APIs for ECUs can be modeled based on I4.0 AdminShell concept
Using Eclipse Vorto to define technology neutral interfaces, supporting C++, c, Java/OSGi

Use Open Group to model ECU landscape and dependencies

e.g. OSGi agent to manage OTA dependencies
THANK YOU FOR YOUR ATTENTION!