IOT: CONNECTED VEHICLE

ARCHITECTURE, CHALLENGE, FUTURE AND OPPORTUNITIES

JIAEN.GUO@BOSCH.COM;
OLIVER.SCHEELE@BOSCH.COM;

BOSCH

About the speakers

1. IoT Architecture & Infrastructure
    1. General infrastructure & architecture
    2. System architecture

2. Challenge
    1. Security
    2. Reliability & performance
    3. Operation
    4. Customer and country oriented data privacy

3. Future & Chance
    1. Branch
    2. Example & demo

Startup inside Bosch for mobile solutions & connected car solutions

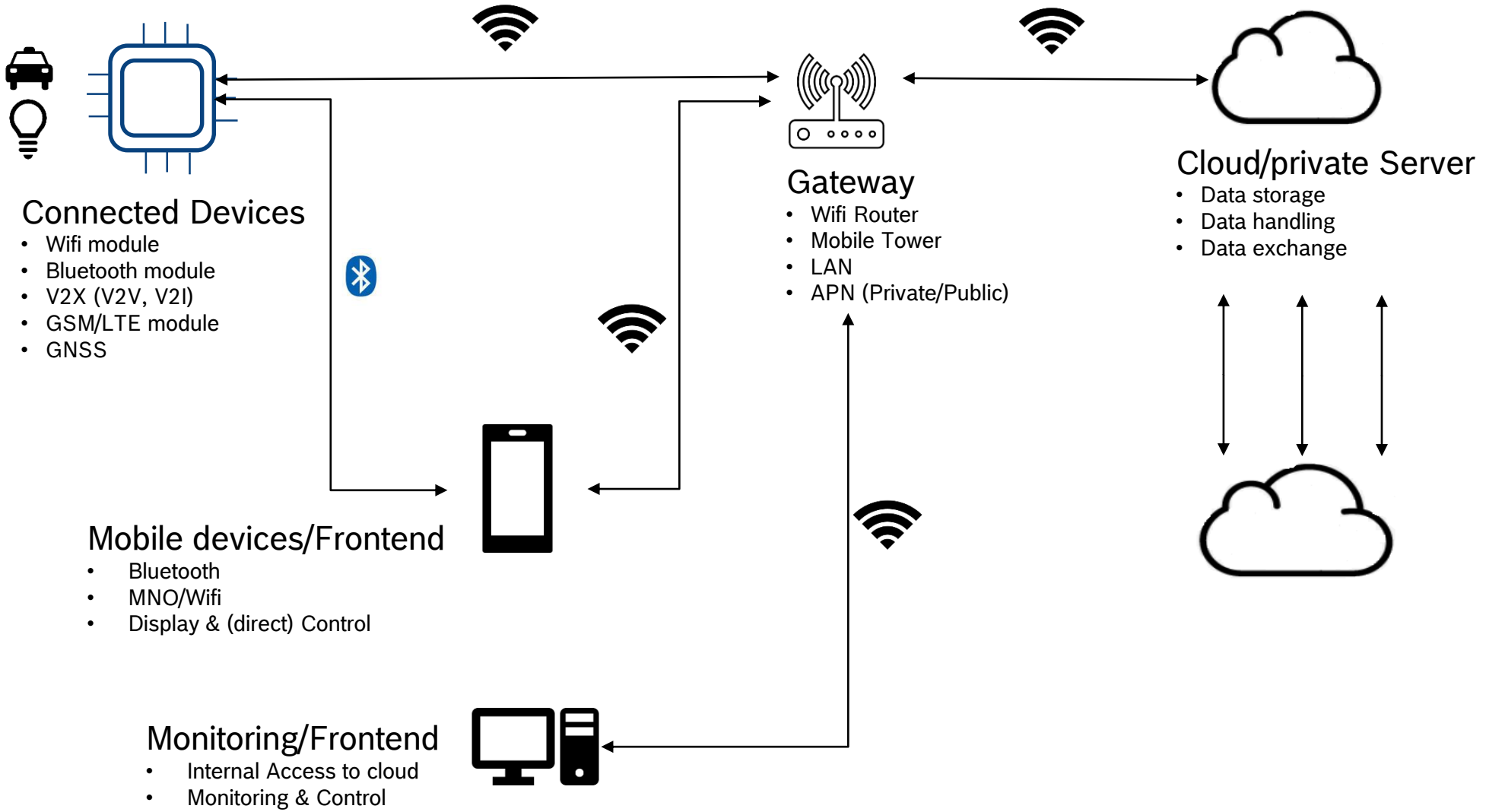
Author: jiaen.guo@bosch.com

Co-Author: oliver.scheele@bosch.com

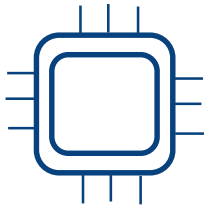
- Leader backend, SW architect.
- Security design/Dev
- Cloud based DevOps

- Group leader & project manager
- Specialist of firmware/hardware
- Overall systems engineering

1.1 General infrastructure & architecture



1.2 System architecture



Connected Devices
(Data generator)
- Impl. of conn (Wifi/BT/MNO)
- Impl. of microcontroller
- Impl. of memory usage
- Impl. of data handling (sensor data, GNSS etc.)
- W/O operation system, e.g. OSGI or direct Firmware programming

Cloud platform or private Server
(Data handling)
- Firewall/DNS/Loadbalancer
- Management platform

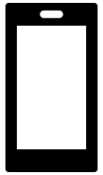
Incl. User Mgt, Application Mgt and Marketplace

- Infrastructure services in Cloud marketplace,

e.g. Oracle, MongoDB, MQ, Email, SMS,

- Data storage & exchange in real time
- Tools for application Mgt (deployment and monitoring)

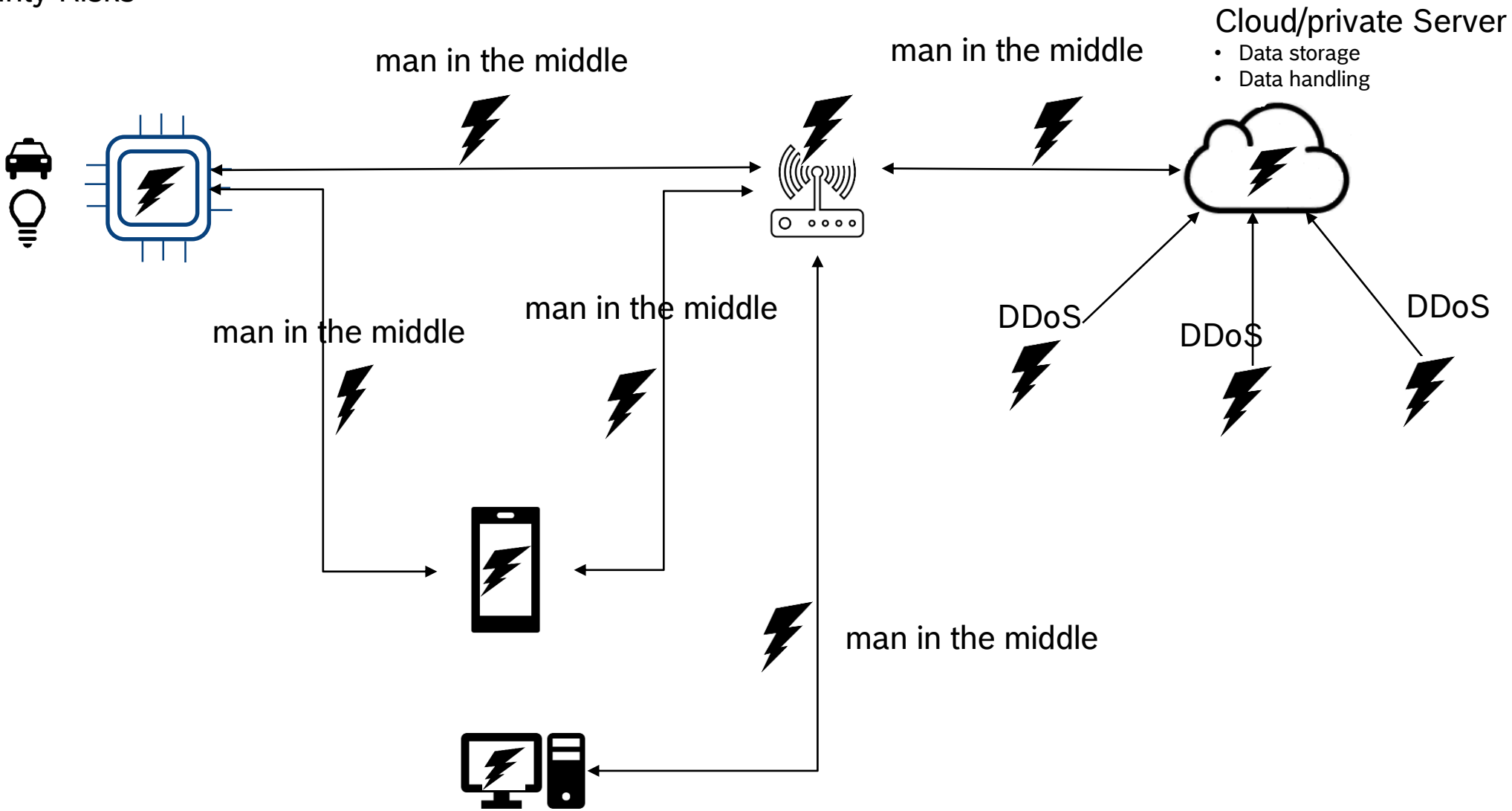
Gateway
(Proxy)
- Data Exchange between device and the backend



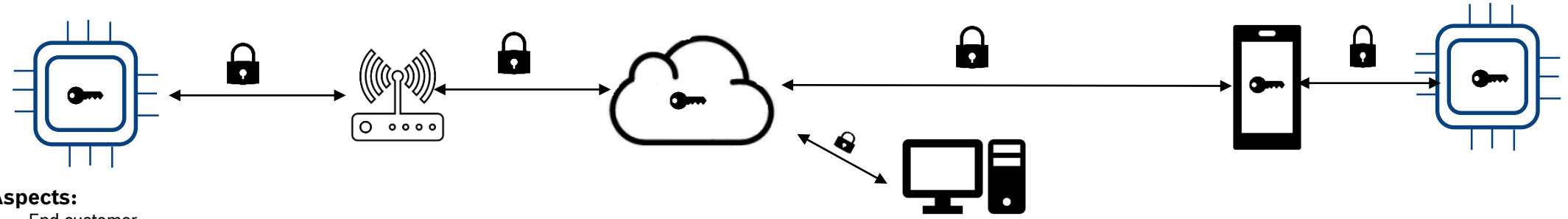
Mobile devices
(Data generator + Access point for Mobile User)
- Data exchange with Device(s) in real time (BT)
- Generation of additional data, e.g. GPS, user data etc.
- Data exchange with backend

Monitoring center
(Access point for supervisor)
- Data exchange with backend
- Monitoring of device data and application data.

2.1 Security Risks



2.1 Security Requirements



Aspects:
- End customer
- Business operator
- Software(incl. Firmware) supplier

General Data security
- Confidentiality(user authentication and authorization)
- Data integrity(data correctness)
- System availability

General key management
- Keep the key safe (physical isolation, nobody can touch it in the runtime)
- Kept in encrypted form
- Dynamic Data/Communication encryption (Symmetric/Asymmetric)
- A key loss may only have local impact, but no global system impact.

Connected Devices
(Data generator)
- Avoid the data manipulation
- Protect of the micro-controller against the hacking of the firmware

Gateway
(Data delegation)
- Keep the communication channel safe
- Nobody can manipulate the data package on the way

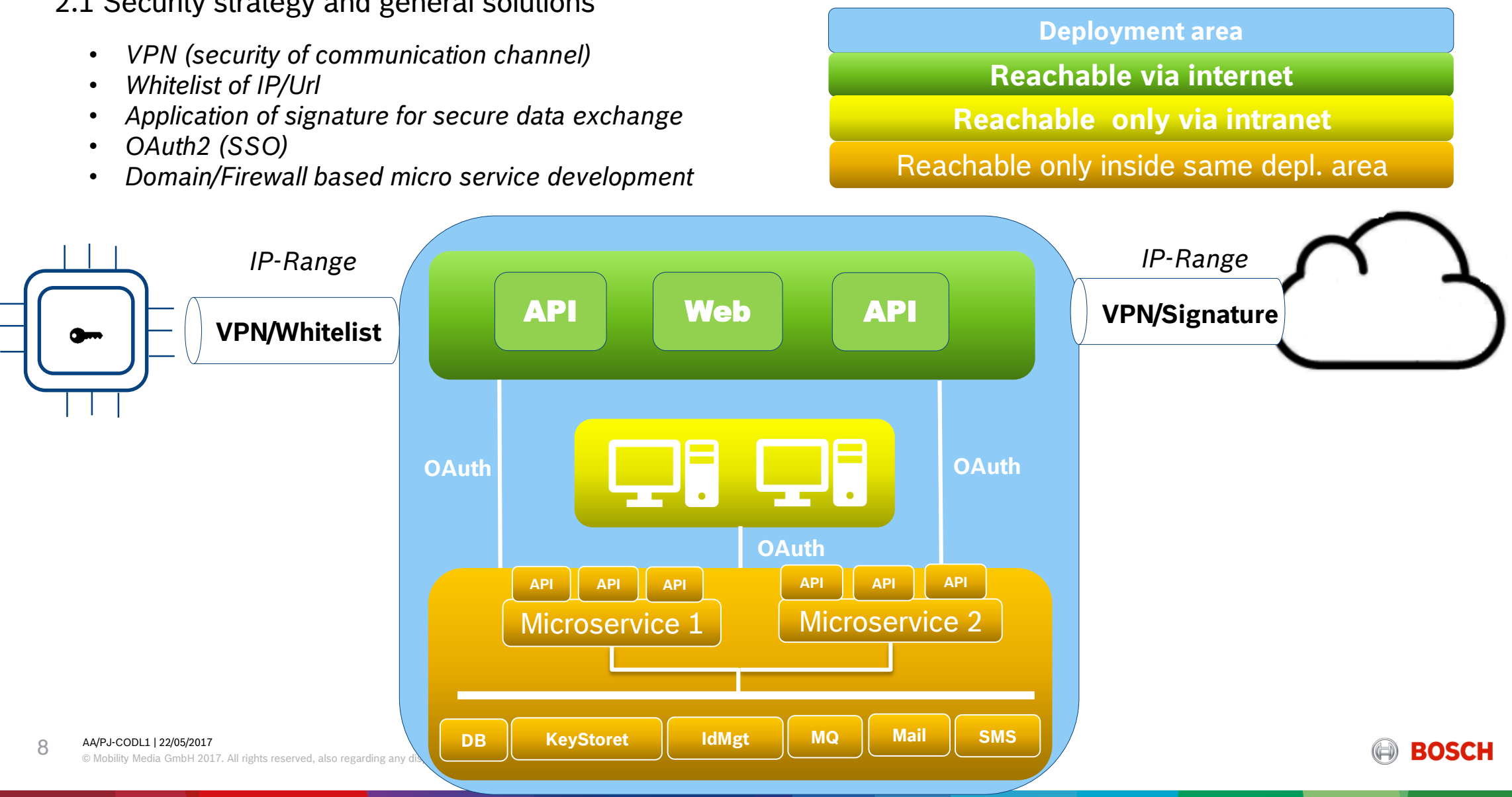
Backend in the Cloud
(Data storage/handling)
- Receive only data from registered devices
- Make sure the received data is not manipulated
- No data lost even the server is overloaded
- Authorized data access

Monitoring center
(Supervisor)
- Authorized data access
- Isolated from Internet by fireware
- Only the meta data is visible

Frontend/Mobile devices
(Data storage/handling)
- Authorized access of personal data

2.1 Security strategy and general solutions

- *VPN (security of communication channel)*
- *Whitelist of IP/Url*
- *Application of signature for secure data exchange*
- *OAuth2 (SSO)*
- *Domain/Firewall based micro service development*



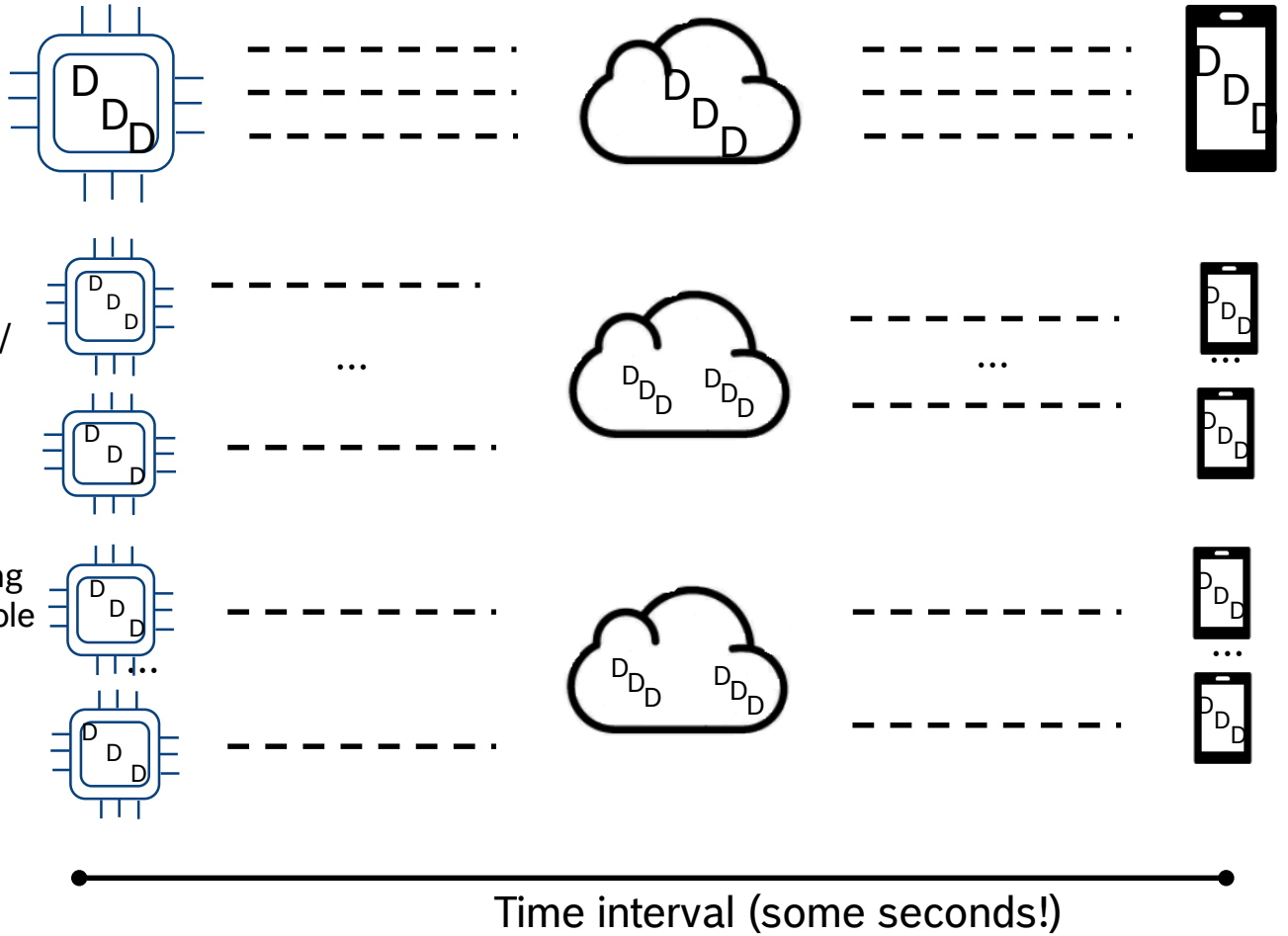
BOSCH

2.2 Reliability & performance requirements

- No data lost
- No unexpected data transformation

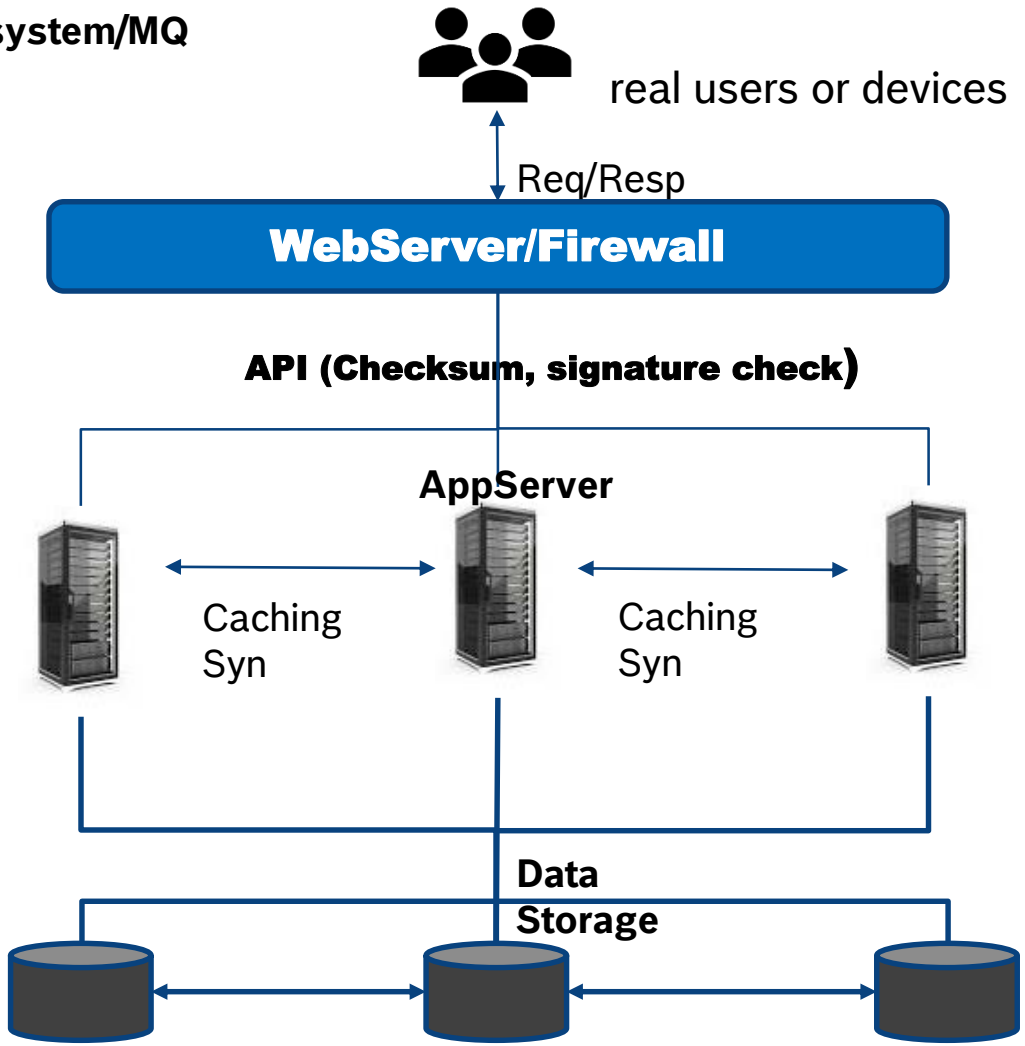
- Reliable transfer of big data / continuous data streaming

- Acceptable data latency during the transfer of big data in the whole system



2.2 Reliability & performance strategy and general solution

A. Cluster Design/Redundant system/MQ



2.2 Reliability & performance strategy and general solution

B. Fault-tolerant programming + data error tracking

Example: Parse jsonData using jsonjackson-databind-xxx.jar

String dataFromIoTDevice: {“DeviceId": “1234567", “utc": 1493212463,

"gps": {"lat": 1.2000, "lng": 2.0000, "alt": 0.0, "sat": 1}, “data_version": “1.2.3.4"}

```
Public String getGPS(String dataFromIoTDevice) {
    ObjectMapper mapper = new ObjectMapper();
        Try {
        JsonNode result = mapper.readTree(dataFromIoTDevice);
        Return result.get(“gps”).toString();
        } catch (catch (JsonProcessingException e) {
            throw new YourException (your own errorMsg, e);
        }
        Return null; }
```

1. Data lost
2. No possibility to track and follow the error!

String dataFromIoTDevice: {“DeviceId": “1234567", “utc": 1493212463,

"gps": {"lat": 1.2000, "lng": 2.0000, "alt": 2.0, "sat": 1}, “data_version": “1.SIM001@<[?]""}

```
Public String getGPS(String dataFromIoTDevice) {
    ...catch (JsonProcessingException e) {
        try{
        mapper.configure(com.fasterxml.jackson.core.JsonParser.Feature.
                                ALLOW_UNQUOTED_CONTROL_CHARS, true);
        JsonNode result = mapper.readTree(dataFromIoTDevice);
        For (JsonNode jsonNode: allJsonChildNodes) {
            if (!StringUtils.isAsciiPrintable(jsonNode.toString())) {
            DBLog.error(jsonNode.key, jsonNode.toString()) )
            }
            Return result.get(“gps”).toString();
        } catch (JsonProcessingException e) {
            DBLog.error(dataFromIoTDevice)
```

Allow all strange characters if the json structure is valid !

Don’t forget to report and follow this behaviour!

2.2 Reliability & performance strategy and general solution

C. Efficient encryption cryptography: Symmetric (ECC) vs Asymmetric (RSA)

	ECC (Elliptic Curve Cryptogr.)	RSA (Rivest-Shamir-Adleman)
Basic Principles	Generate keys through properties of ellipic curve equation	Factor the product of two large primes
Key/signature length for the same sec level of 80 bits $(2)^{80}$	Key: 160 bits Signature: 320 bits	Key 1,024 bits Signature: 320 bits
Signature benchmark (openssl 1.0.2 beta on x86_64): sign/s	ECDSA using key of 256 bits: 9516/s	RSA using 2048 bits key: 1001/s
En-/De-cryption	Device individual en-/de-cryption for req and resp.	Encryption of Device Resp only based on server pk.
Break of encryption	Need device key and server key	Only need one key

More efficient!

More powerful!

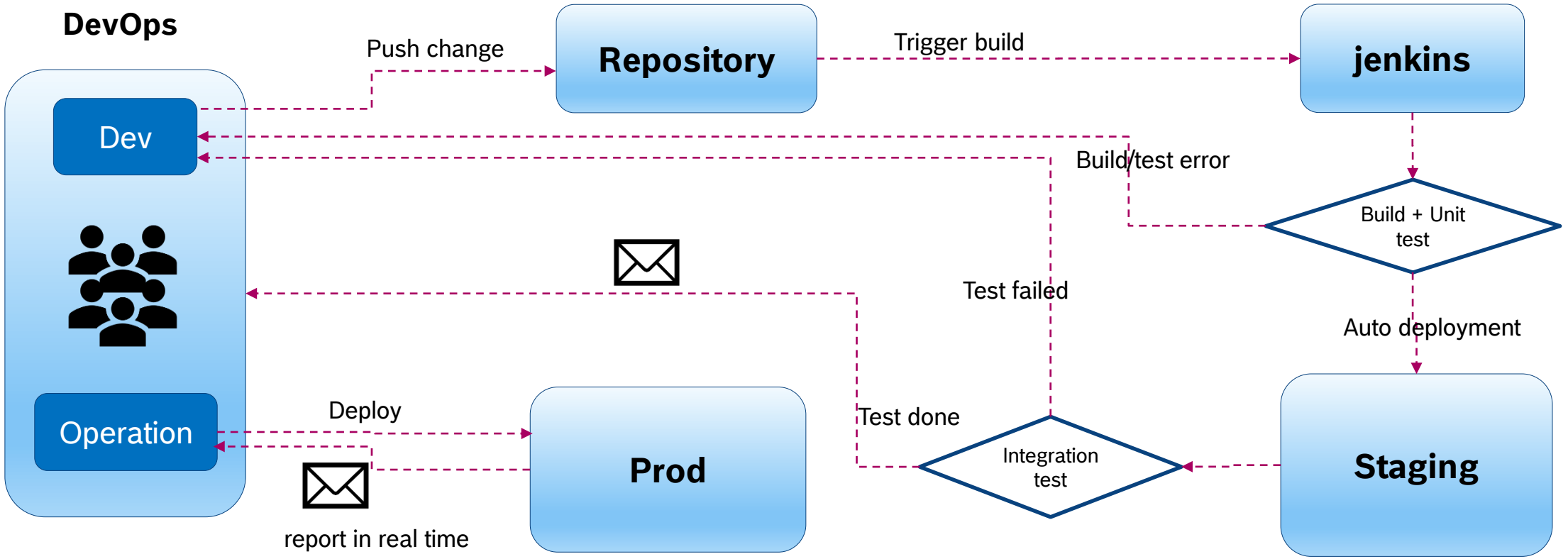
More secure!

2.3 Operation

Mode: DevOps

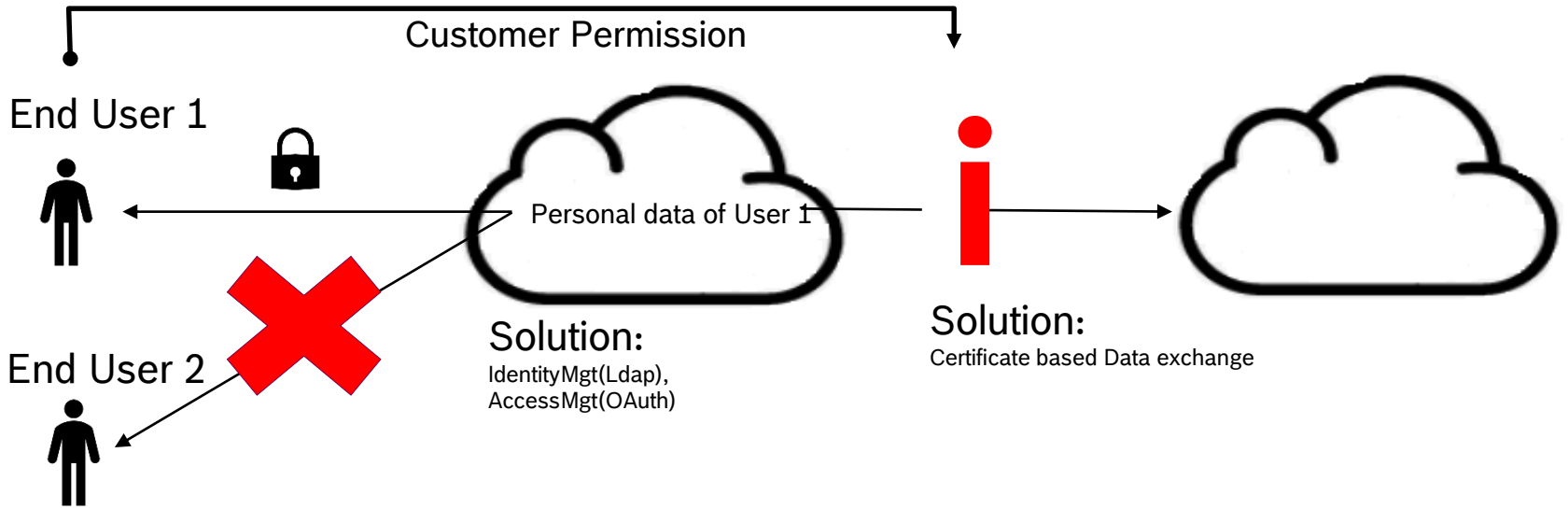
Targets:

- Keep the development reliable & safe
- Quick response (not only reaction!) for questions from customers

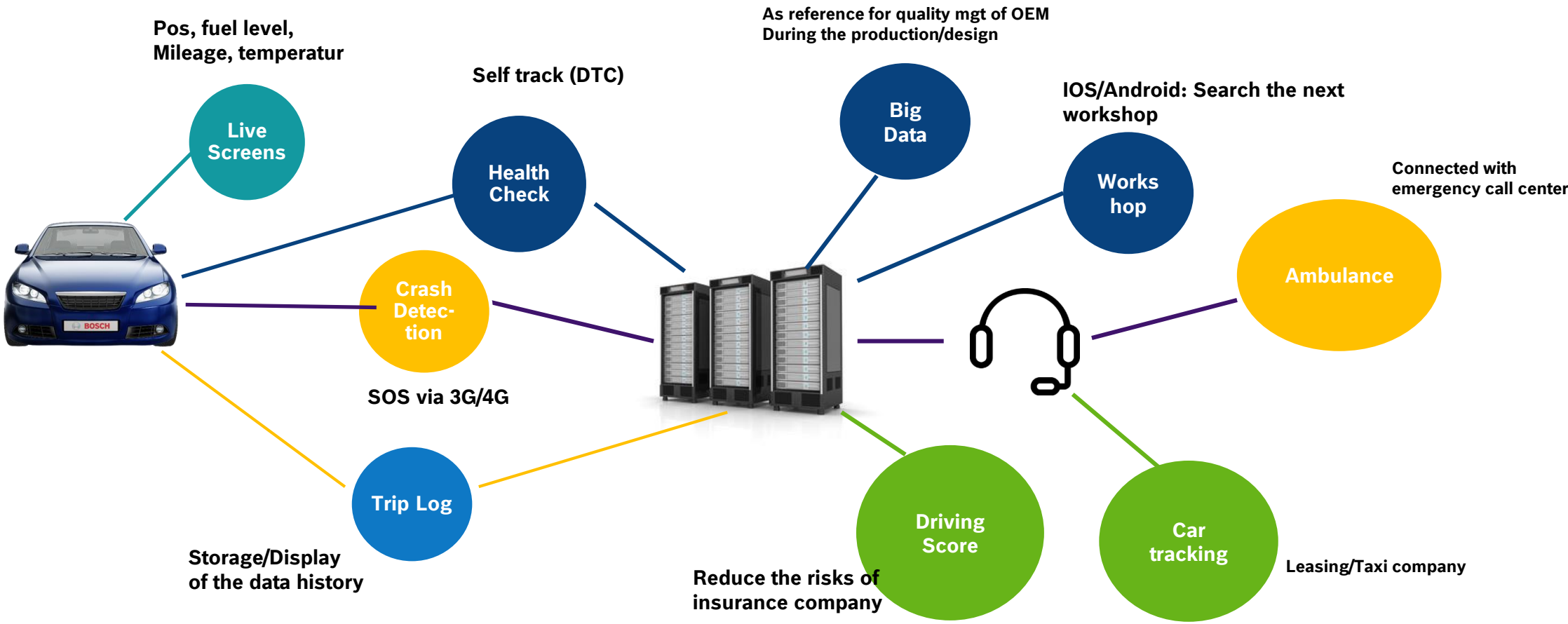


2.4 Customer and country oriented data privacy

- Anonymous vs identified personal data
- Which data belong to whom ?
- How long can customer/device data be kept ?
- Where should which data be kept?
- Special data policy on the country level?



Use Case examples of connected vehicle



3 Project Example: Vehicle Link 3100

- Robert Bosch project with focus on retrofit connectivity
- Data based on OBD II and Bosch sensors (acceleration, gyro, GNSS, thermometer etc.)
- Communication based on Bluetooth and Cellular network (GSM/UMTS)
- NACL as powerful encryption and authentication library
- Cloud based backend development (Bosch IoT cloud, JEE, Microservices)