

# Never again offline?!?

Experiences on the outstanding role of data in a large-scale mobile app ecosystem

Dr. Matthias Naab, Fraunhofer IESE  
Dr. Ralf Carbon, John Deere ETIC  
Susanne Braun, Fraunhofer IESE

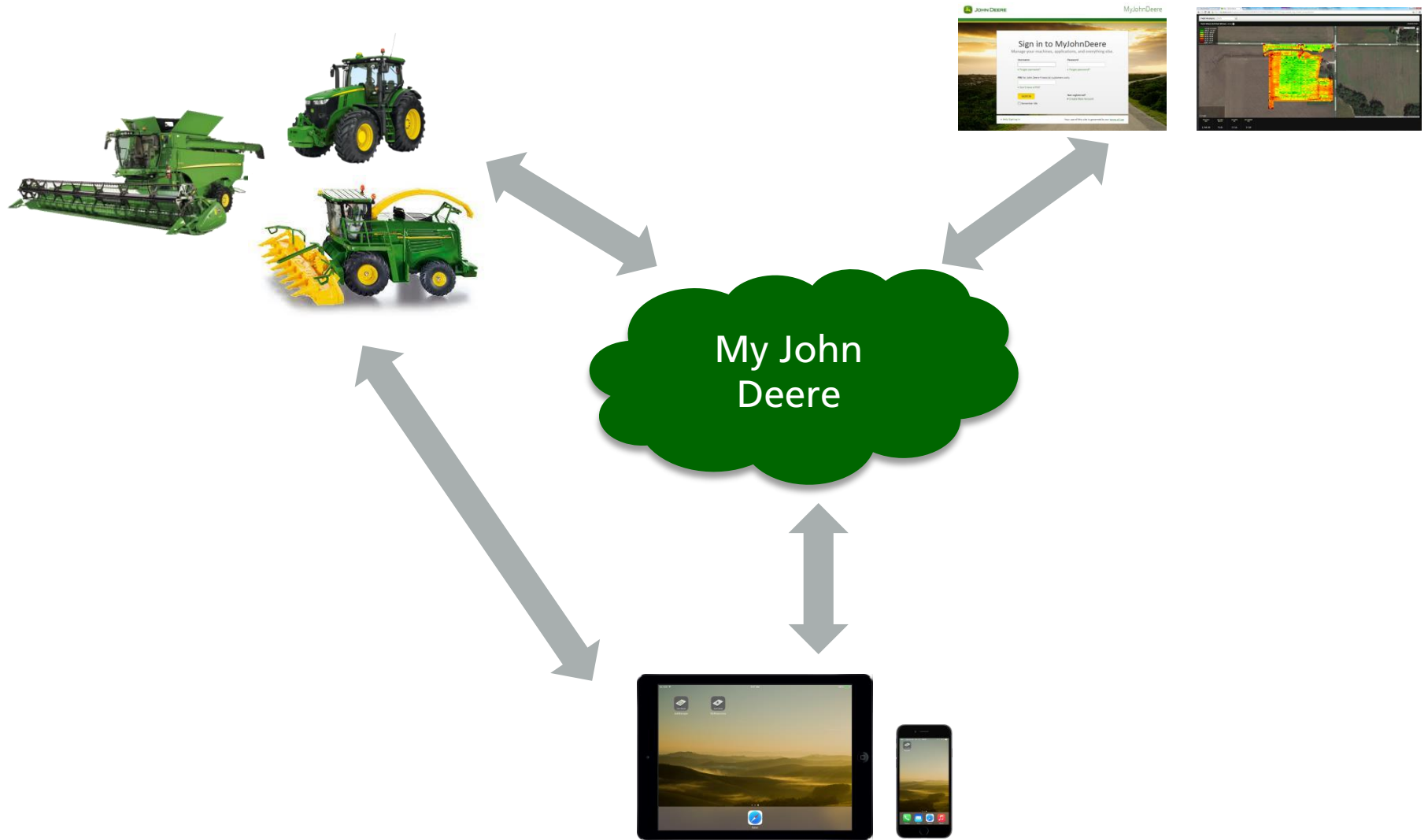
April 29, 2015  
SATURN 2015, Baltimore, USA



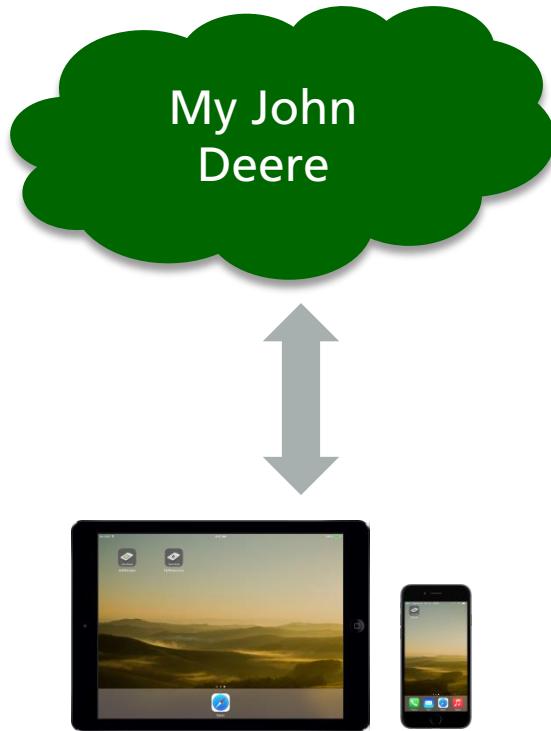




# John Deere Ecosystem



# Introducing Mobile to the John Deere Ecosystem



- One app vs. many apps
- Native vs. hybrid or web
- Online vs. Offline
- Proprietary vs. standard data
- ...

# Collaboration Project

- Duration: since Q4 2013
  - Several phases and scope extensions
- Team
  - ~ 6 full time persons
- Approach
  - UX / Arch / Dev / QA
  - 2 week iterations
  - Testing with real end customers since summer 2014
  - Continuously feeding concepts into production version
  - Re-newing the code base after ~6 months

# Where to find Data in a Typical Architecture Document?

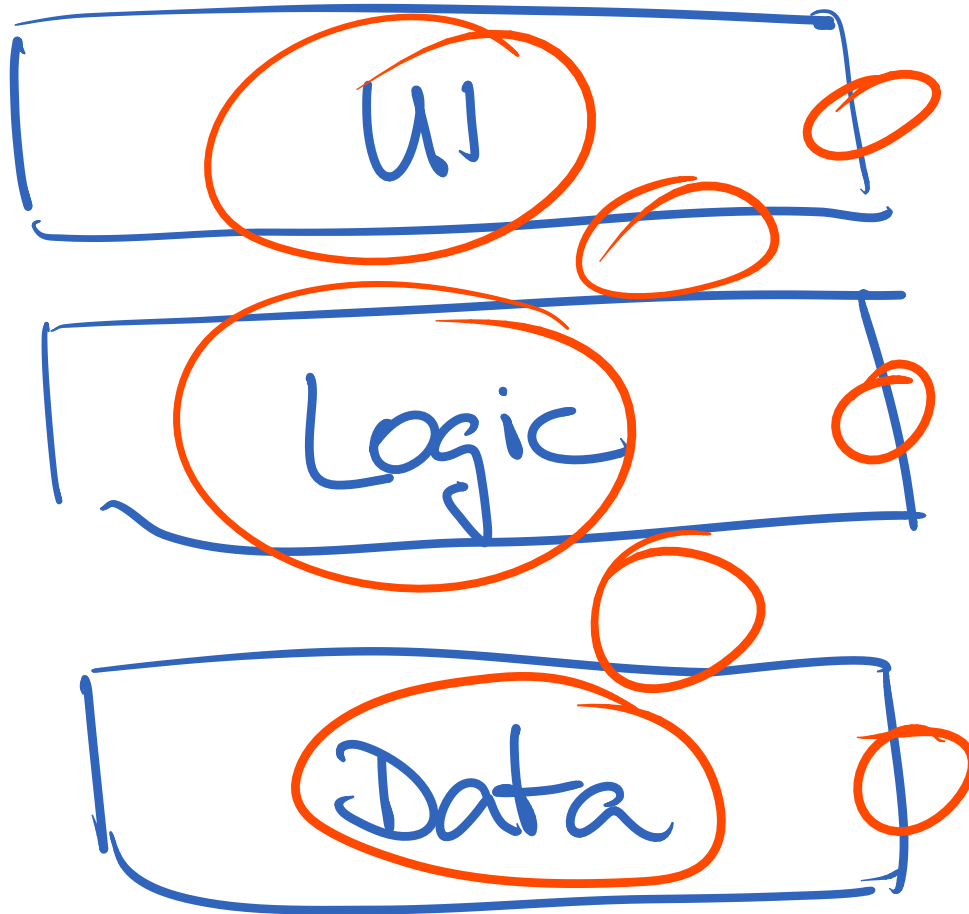
UI

Logic

Data

← Here!

# Where is Data Really???



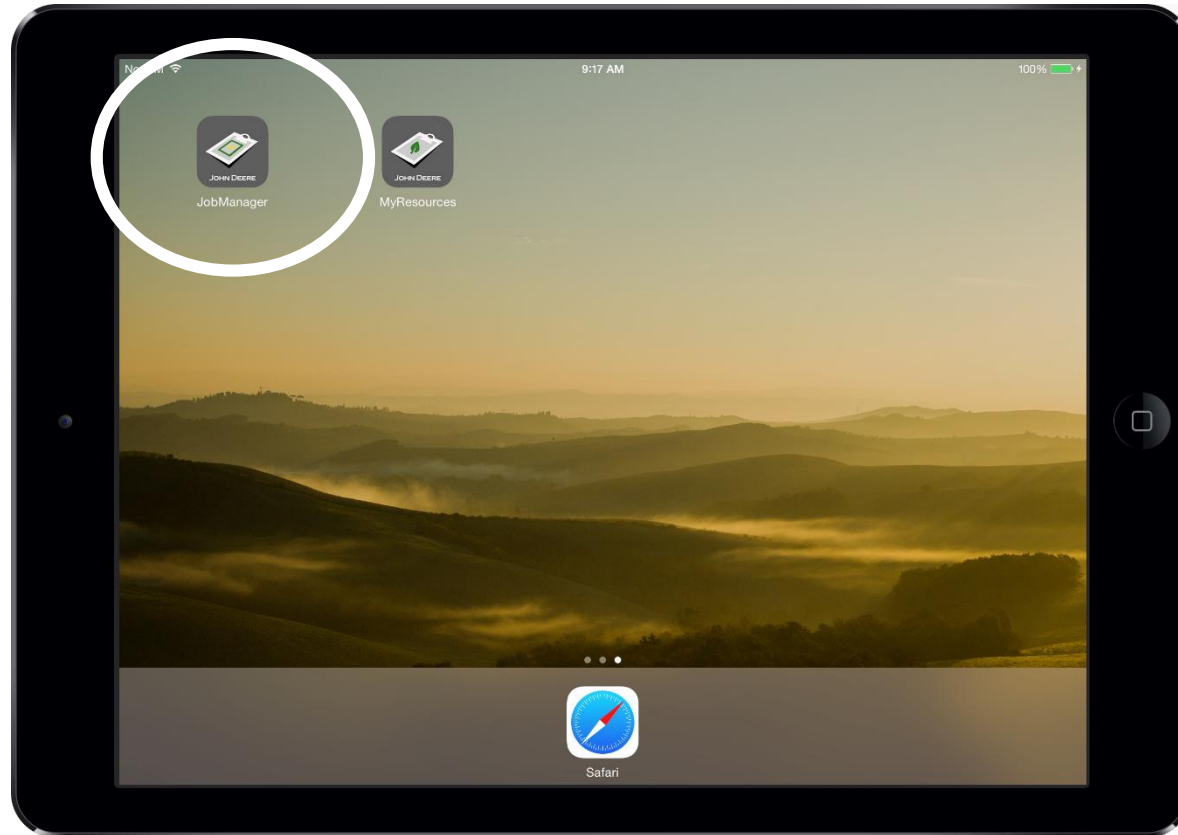
Everywhere !



# What does this mean for architecting?

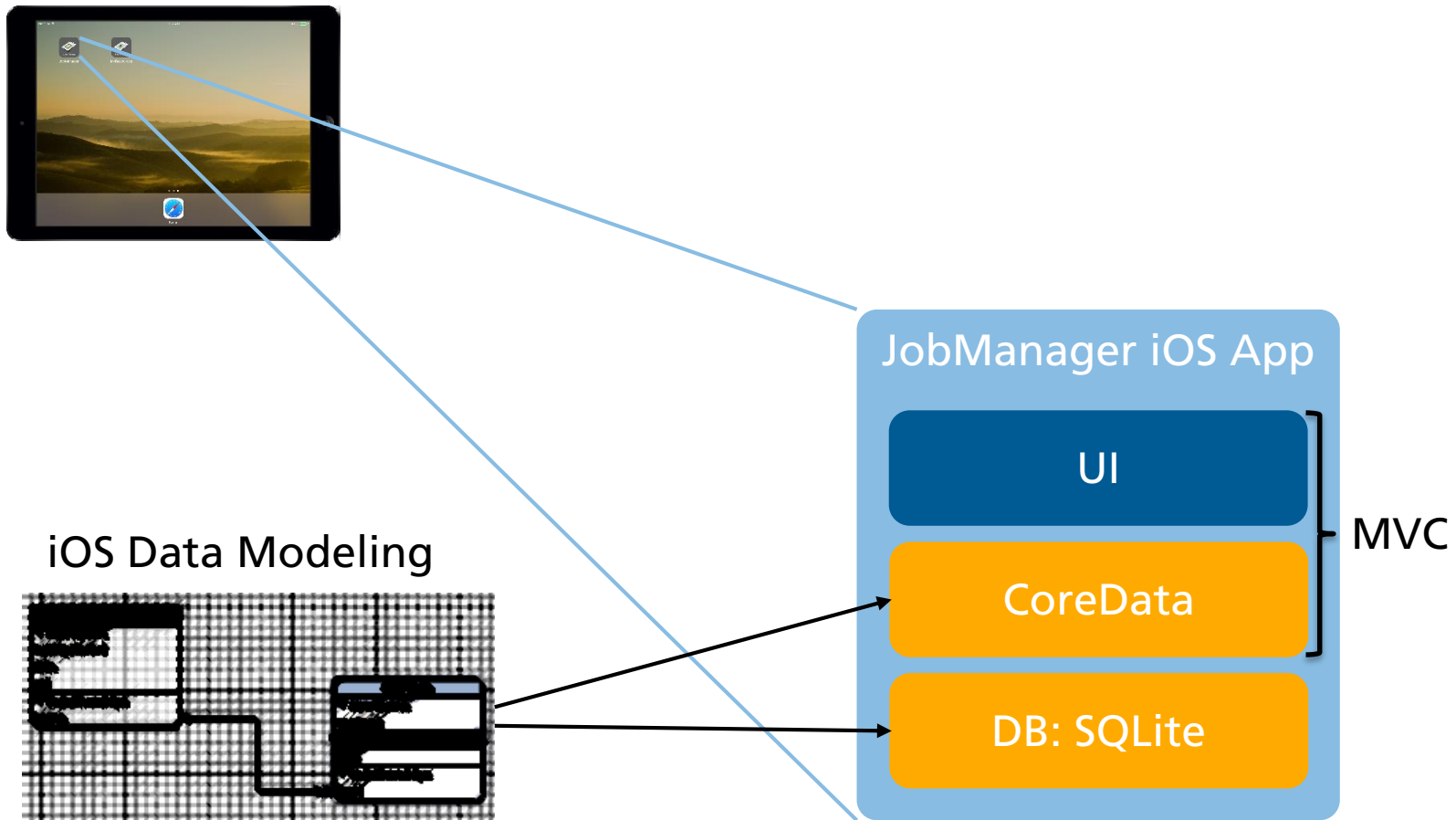
# We need an App ... Architecting it should be Simple!

1



# We need an App ... Architecting it should be Simple!

1



# Multiple Users Work on Shared Enterprise Data

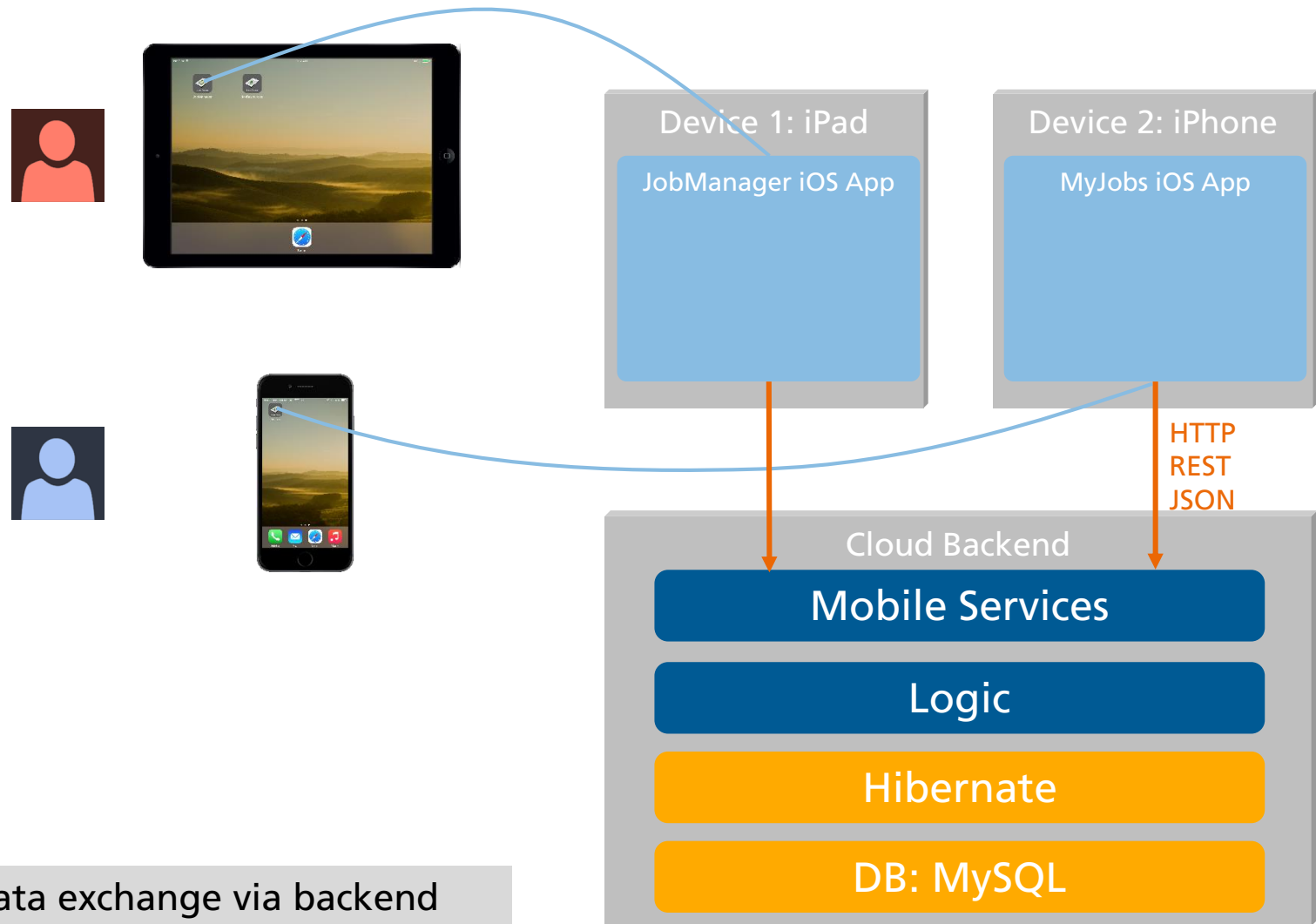
2





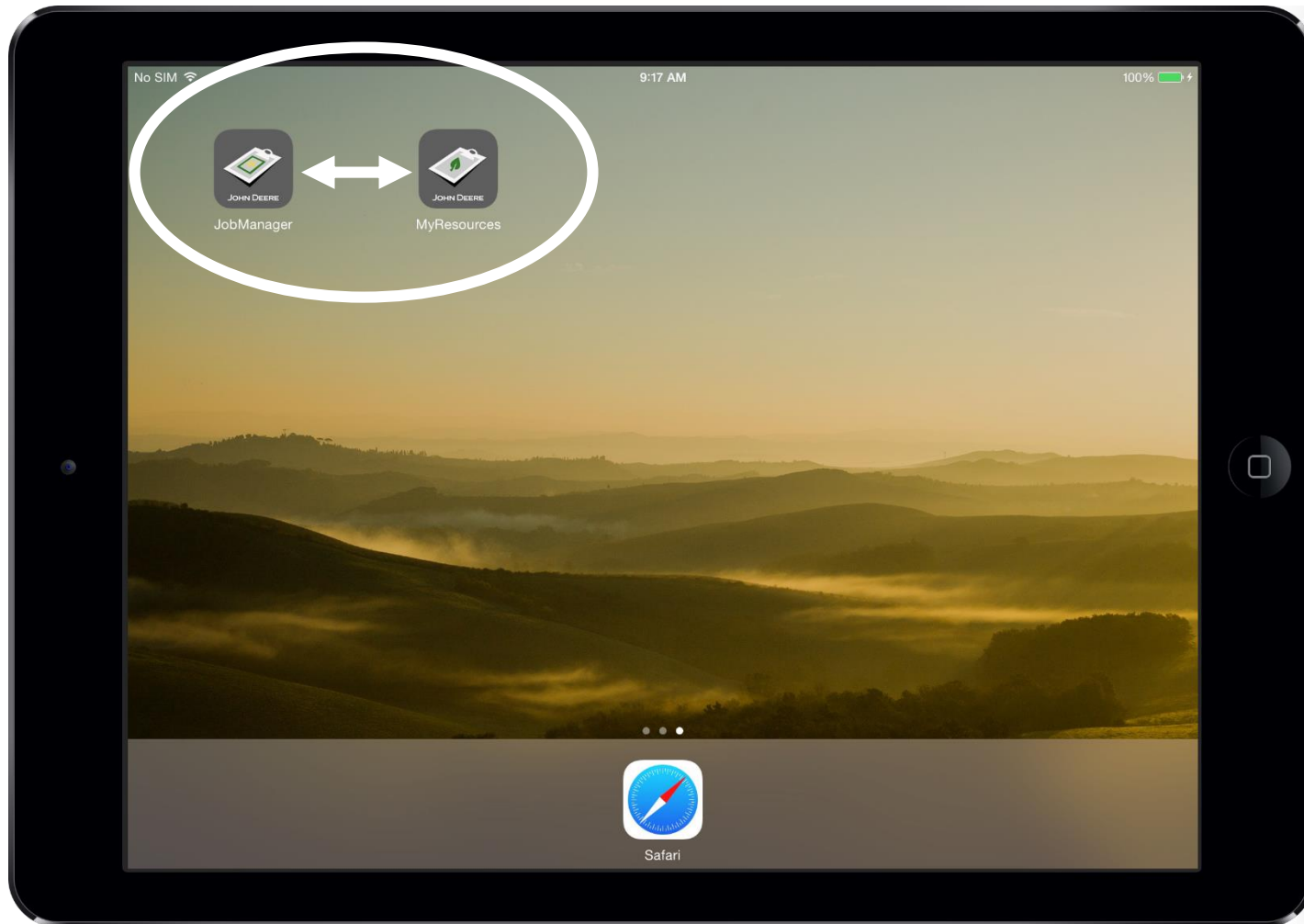
# Multiple Users Work on Shared Enterprise Data

2



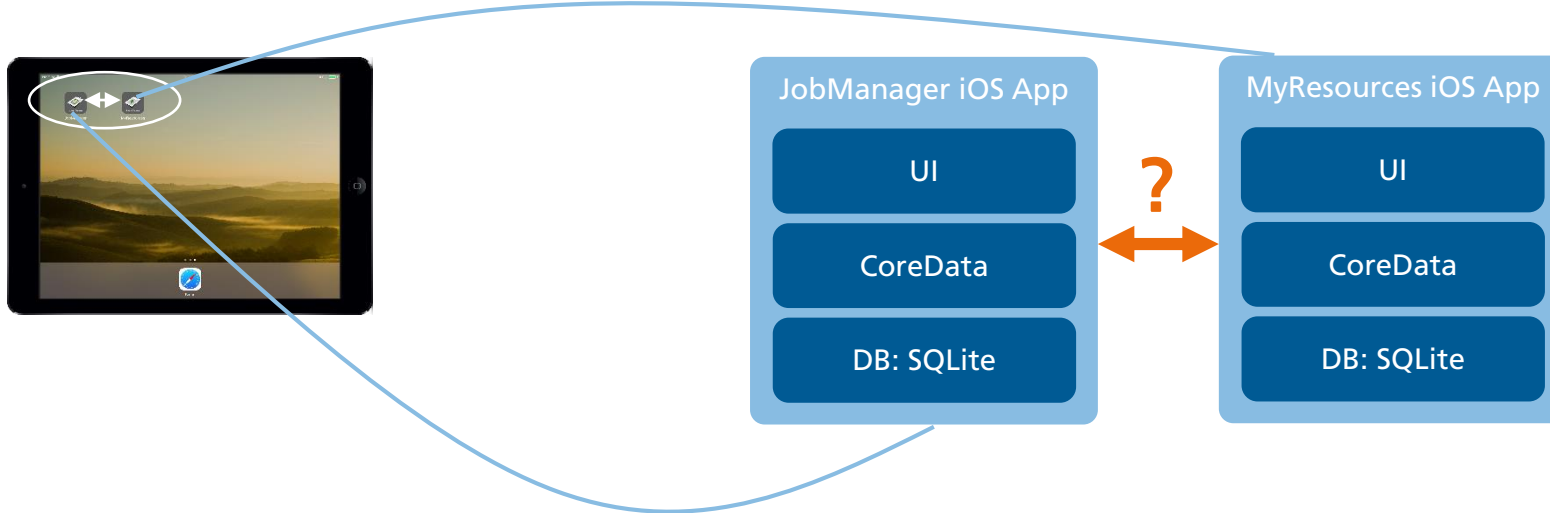
# Apps Work on Shared Data

3



# Apps Work on Shared Data

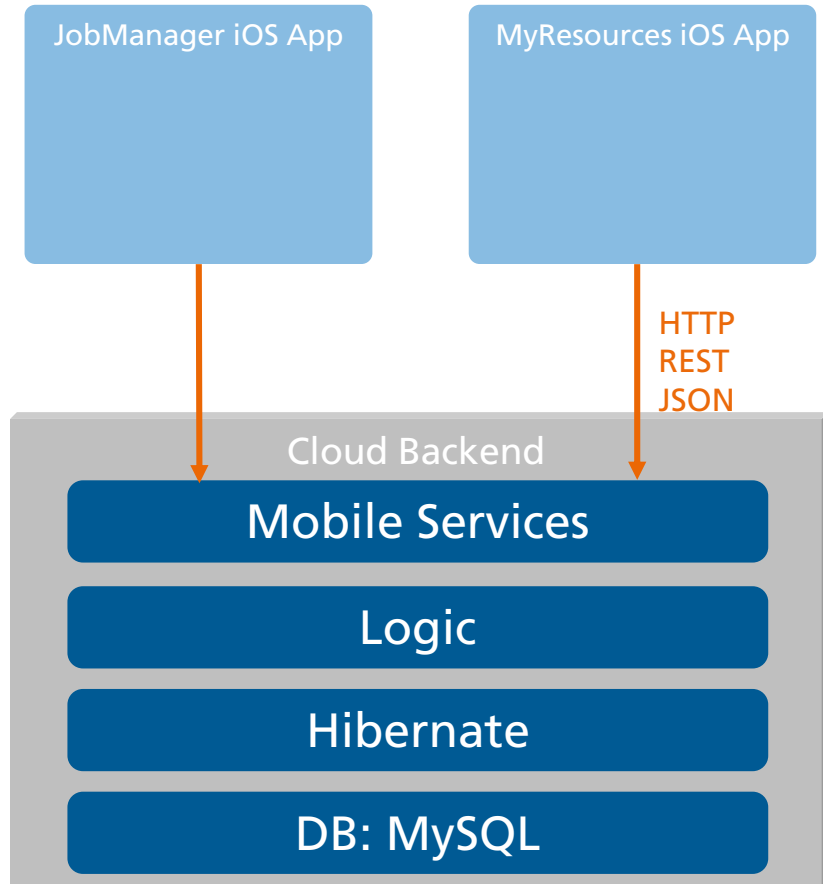
3



- iOS  $\leq 7$  : No shared data

# Apps Work on Shared Data

3



- Data exchange via backend
- On app switch, update data



# Users Have to Work Offline

4



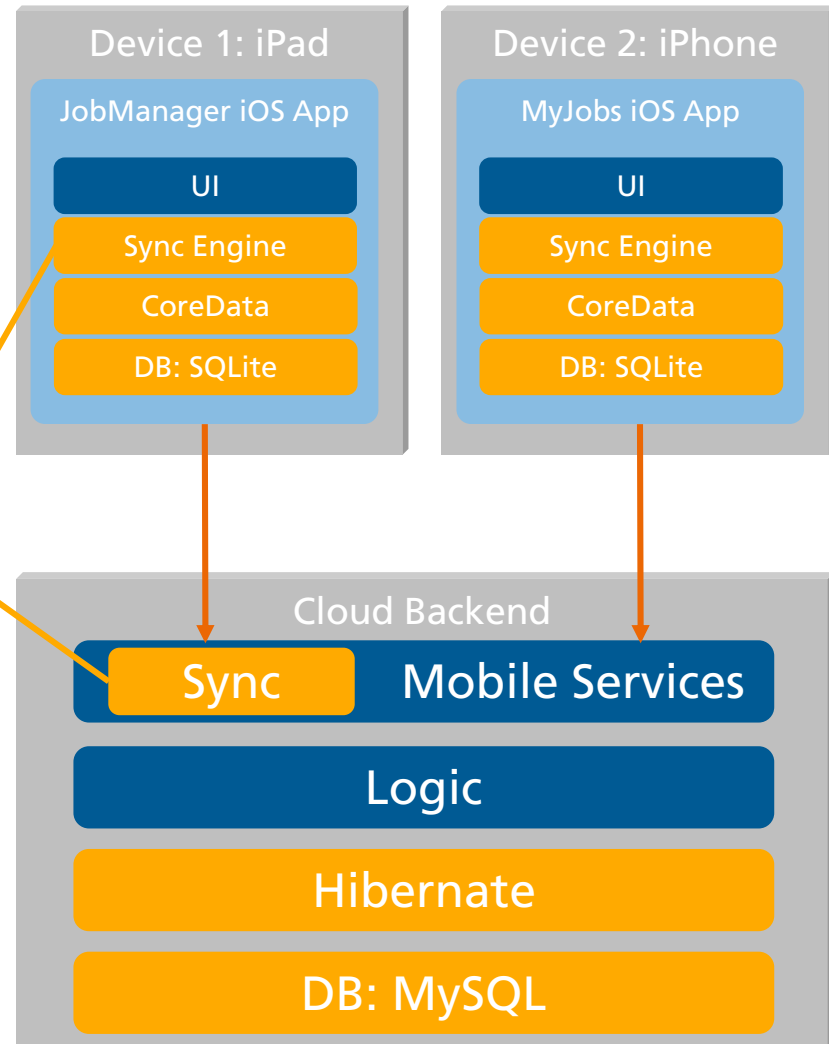
# Users Have to Work Offline

4



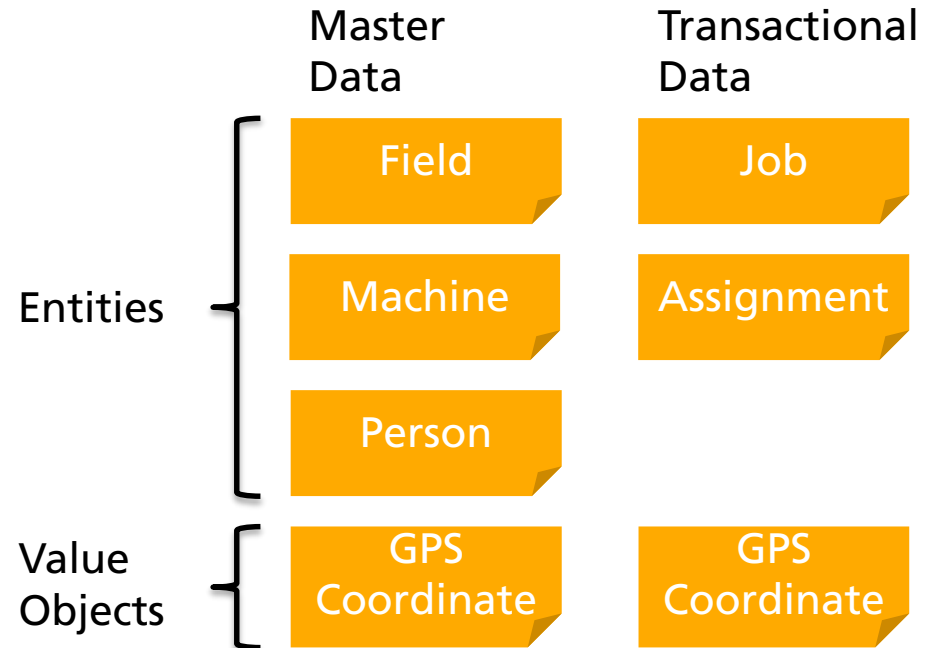
```
job {  
  "id": "UUID",  
  "revNr": "42",  
  "syncState": "UPDATED"  
}
```

- Custom sync mechanism
- Sync service and engine
- UUID + RevisionID + SyncState
- Conflict detection + resolution
- Treatment of deletion
- → High effort!



# Data has Different Characteristics

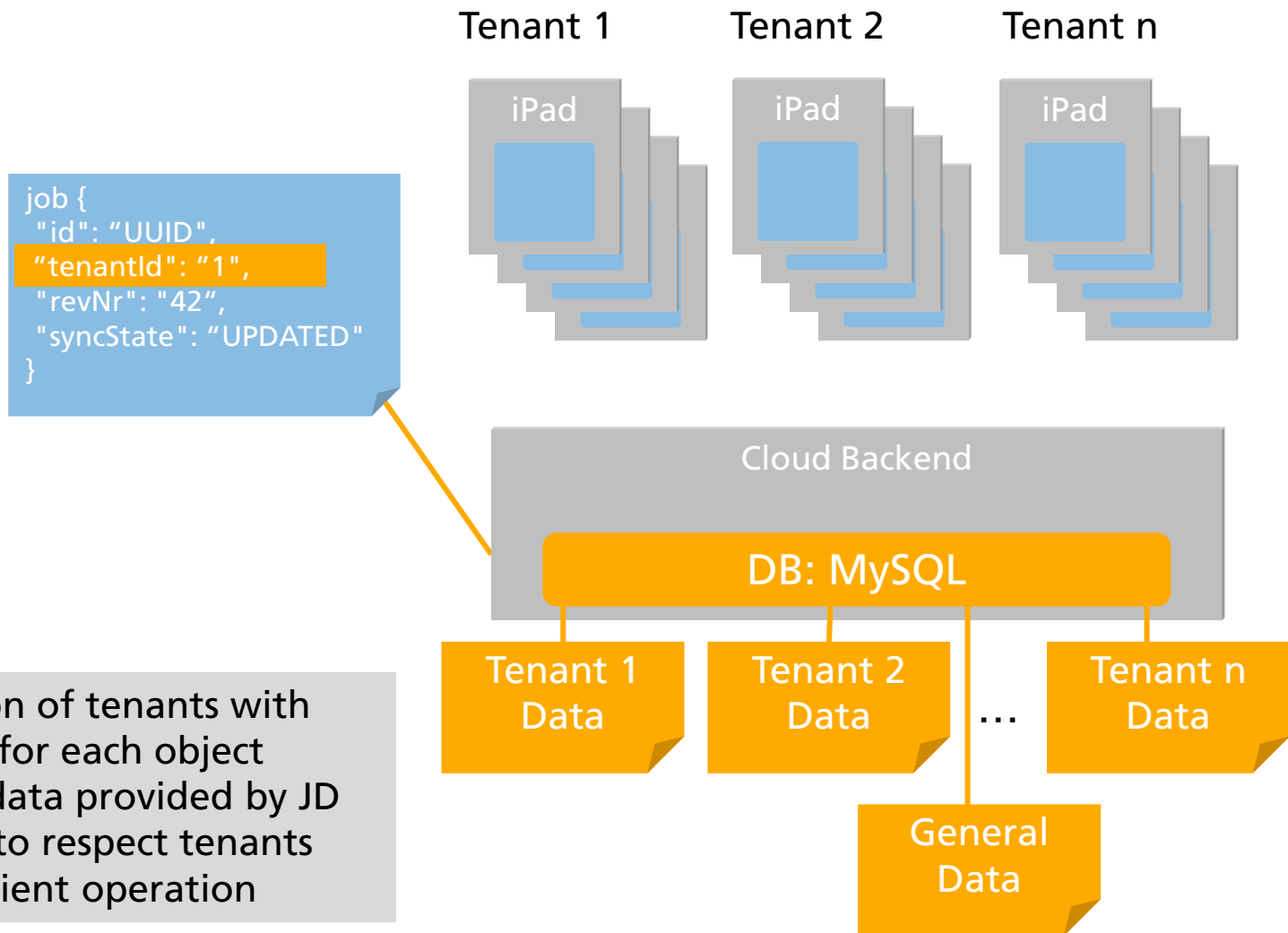
5



- Data classification
- Data modeling rules (e.g. transactional accesses master, usage of aggregates)
- Life-cycle of objects (deletion)

# JD Operates Multiple Tenants on the Backend

6

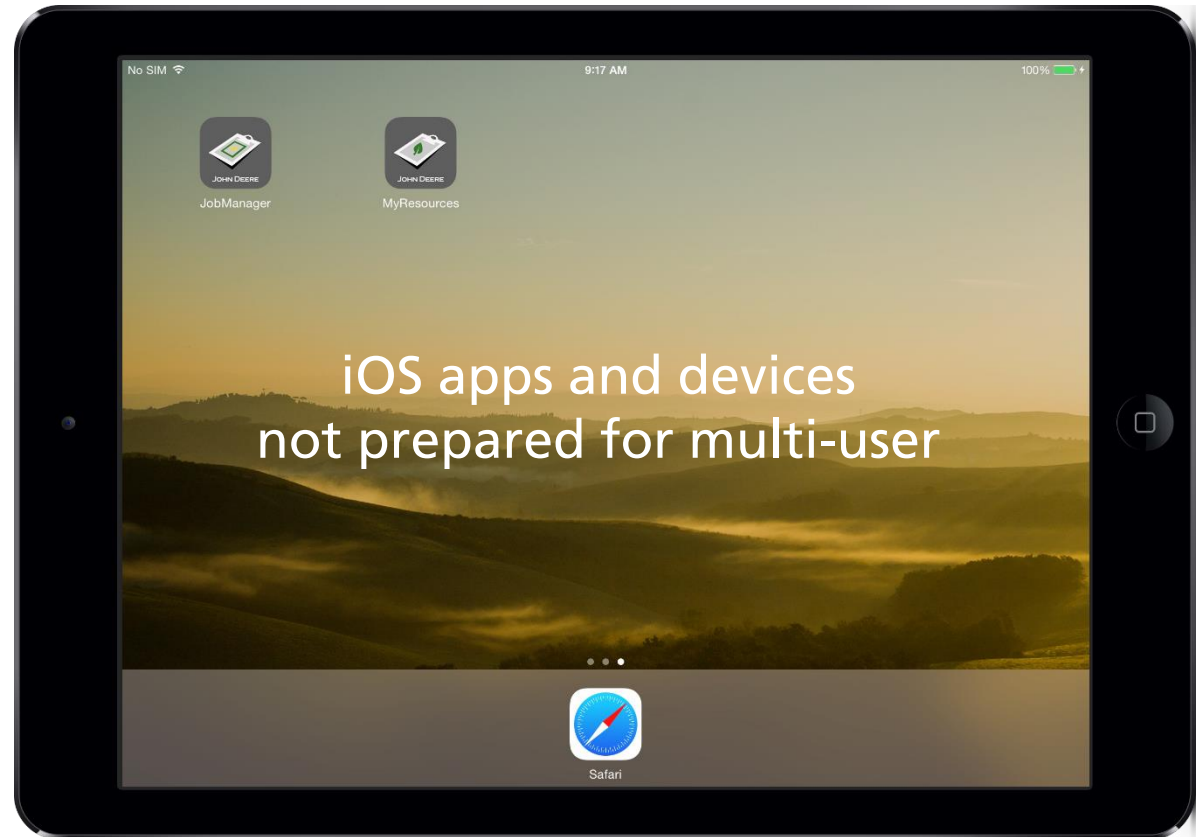


- Separation of tenants with tenantId for each object
- General data provided by JD
- Sync has to respect tenants
- Cost-efficient operation



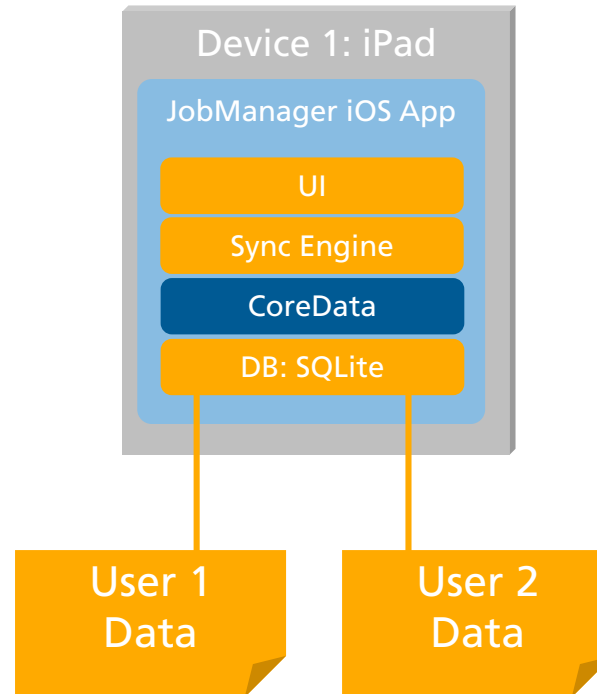
# Multiple Users Share one Device

7



# Multiple Users Share one Device

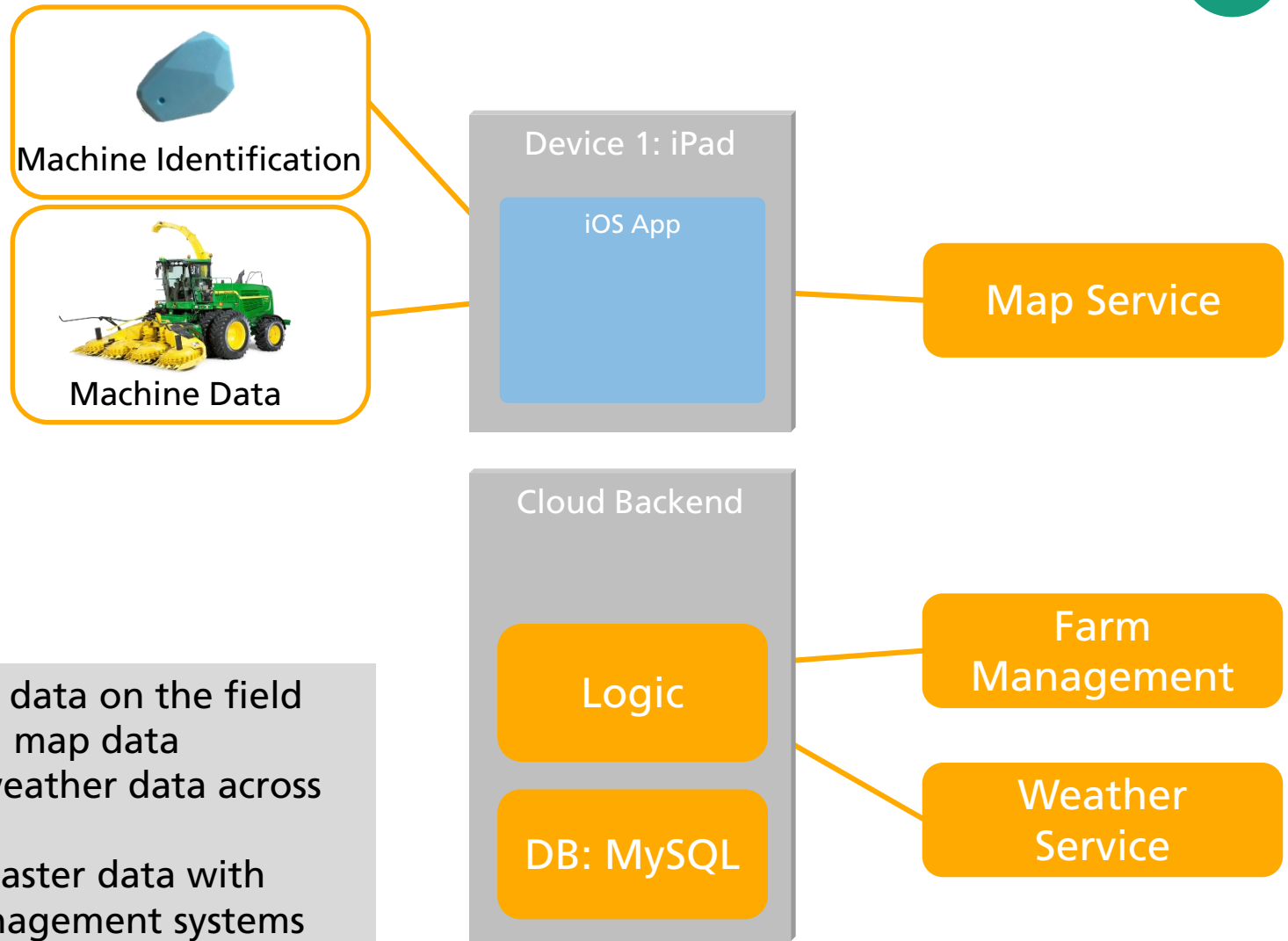
7



- Login mechanism for users
- Custom realization of SSO
- Usage of keychain group access for SSO tokens
- Sharing among apps via backend
- Removing data of logged out users from current SQLite DB

# Connecting to the World

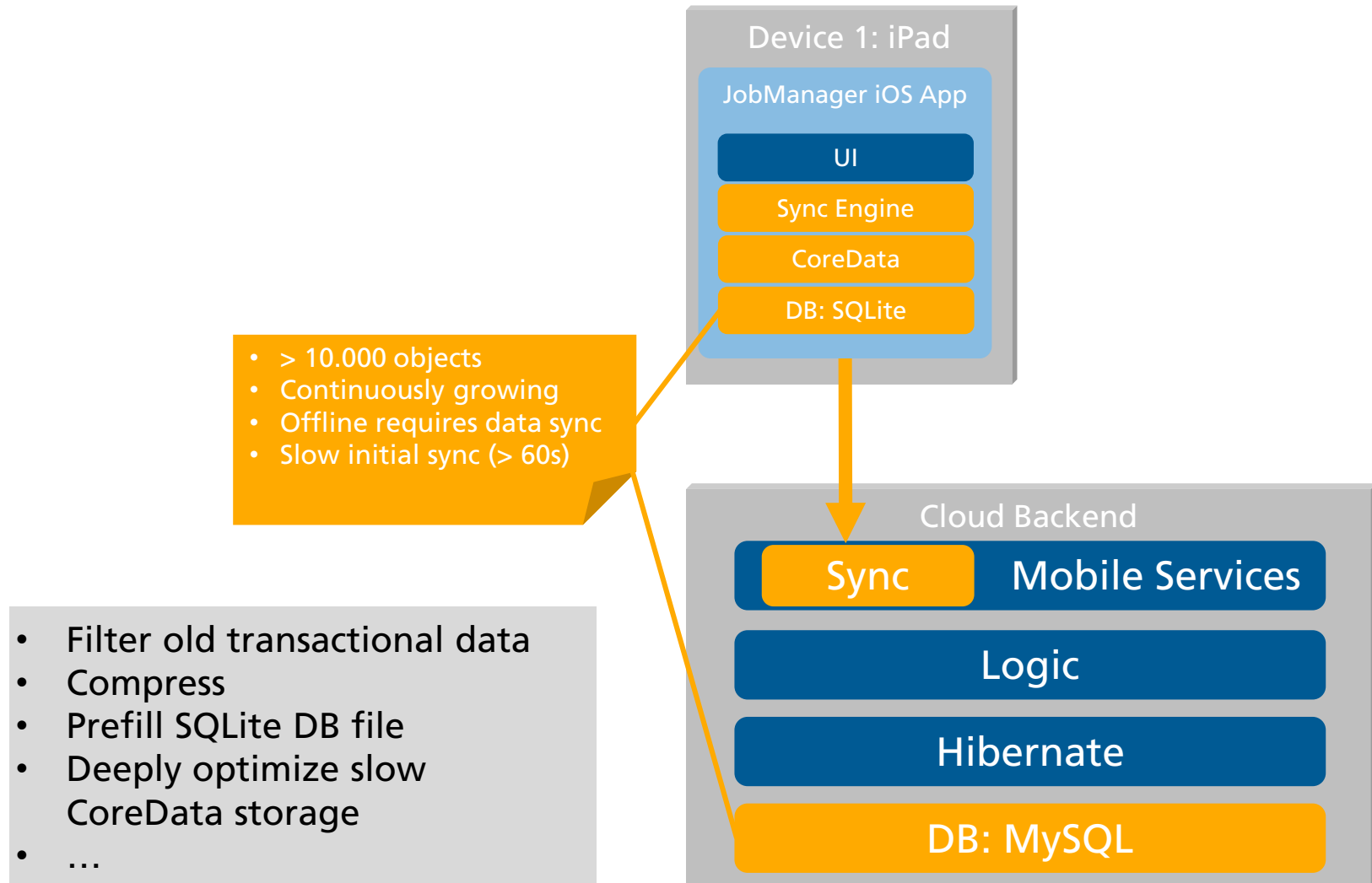
8



- Collecting data on the field
- Streaming map data
- Reusing weather data across tenants
- Sharing master data with other management systems

# Increasing Amount of Data

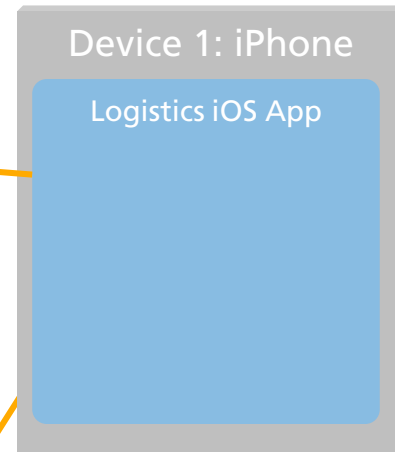
9



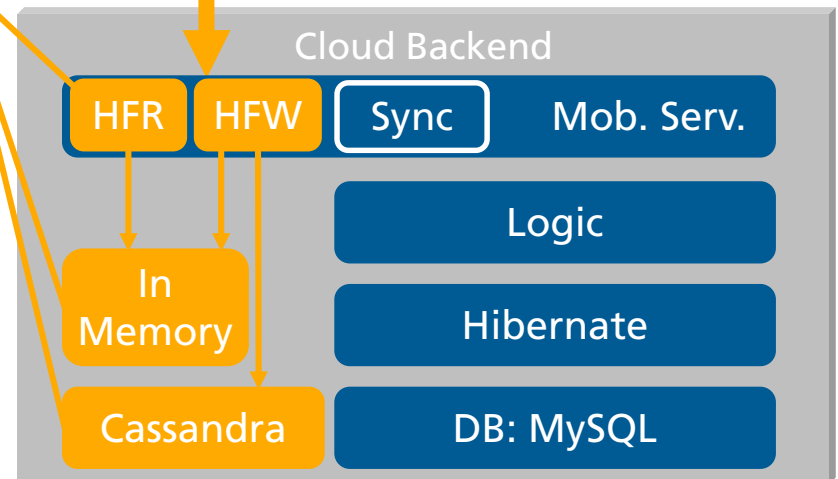


# Increasing Frequency of Changes of Data

10



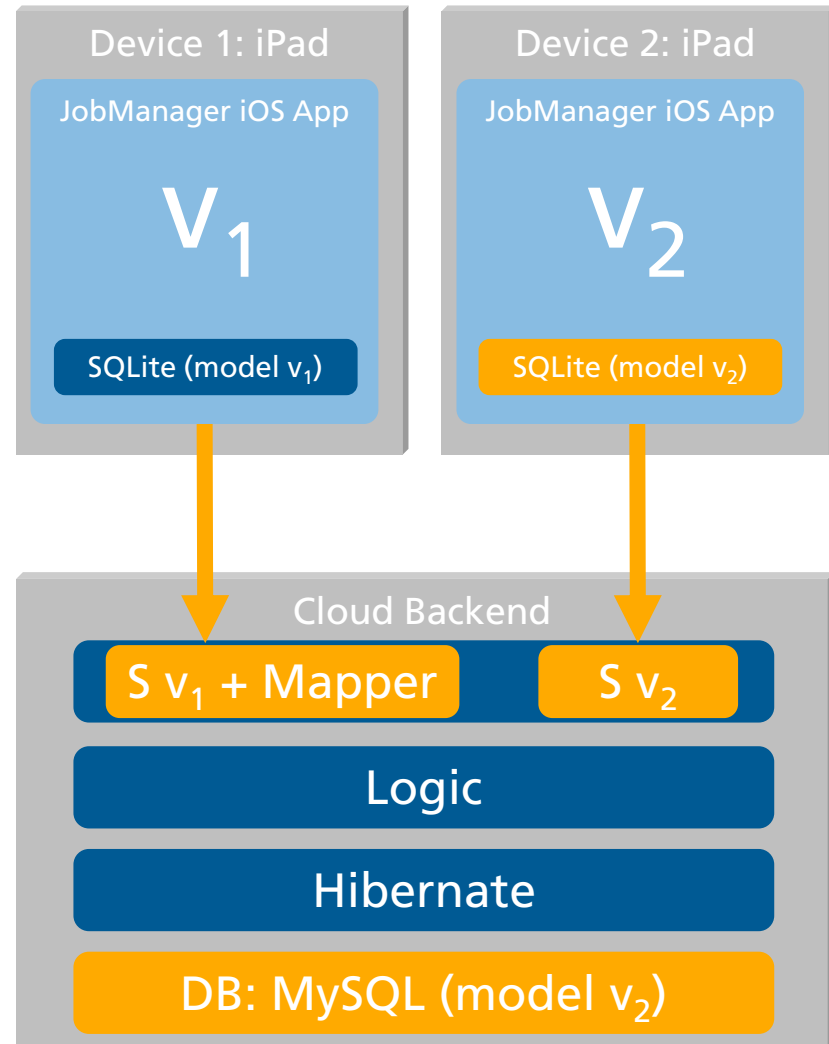
Near real-time data (GPS,...)  
→ ~1 change/s  
Live distribution to devices  
Visualization and analysis



- CQRS: Separate interface for high frequency data
- Event-Sourcing
- Cassandra for fast writes
- In-memory for fast reads

# Upgrading Apps

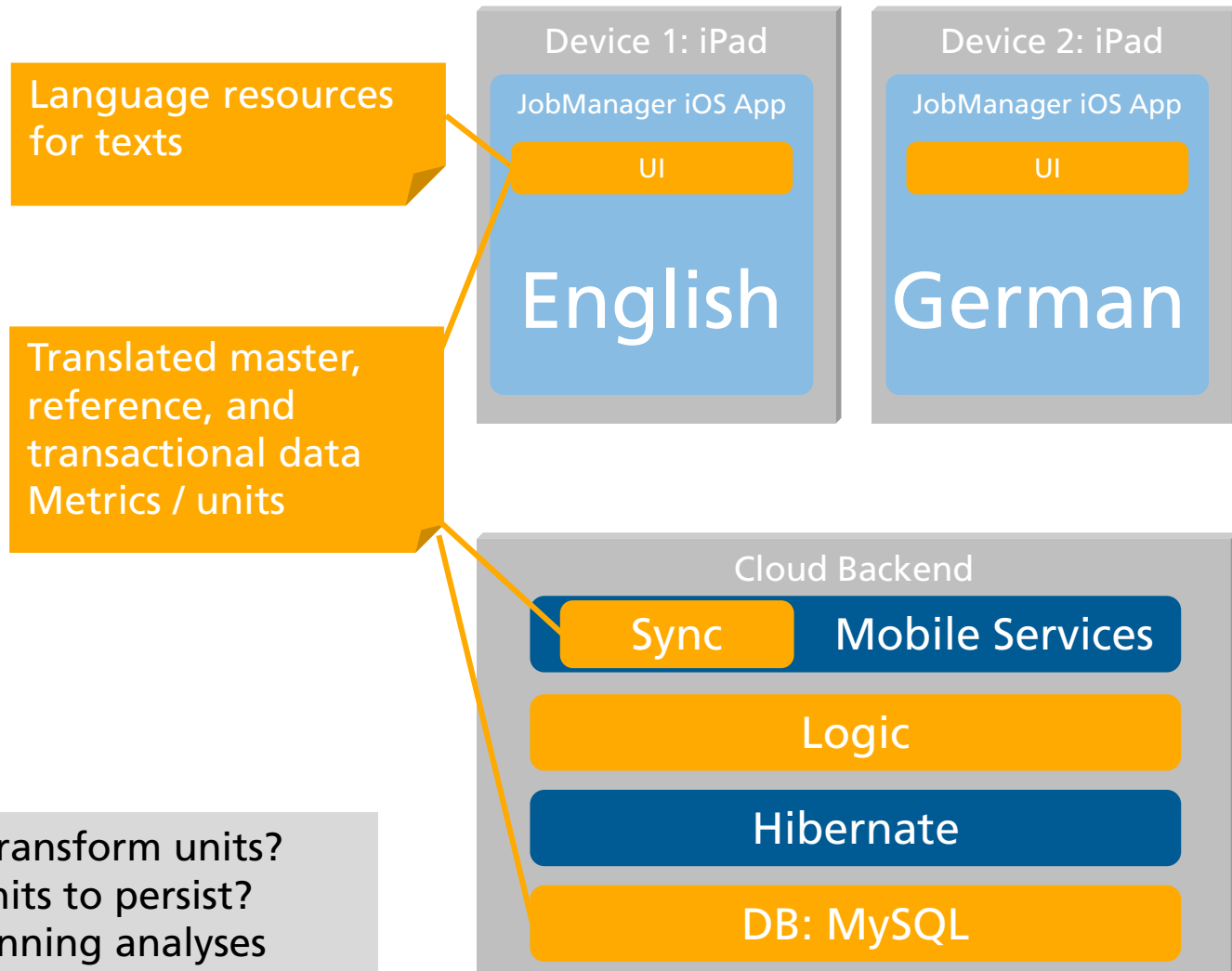
11



- Support multiple app versions
- Maintain single DB model
- Create mapper

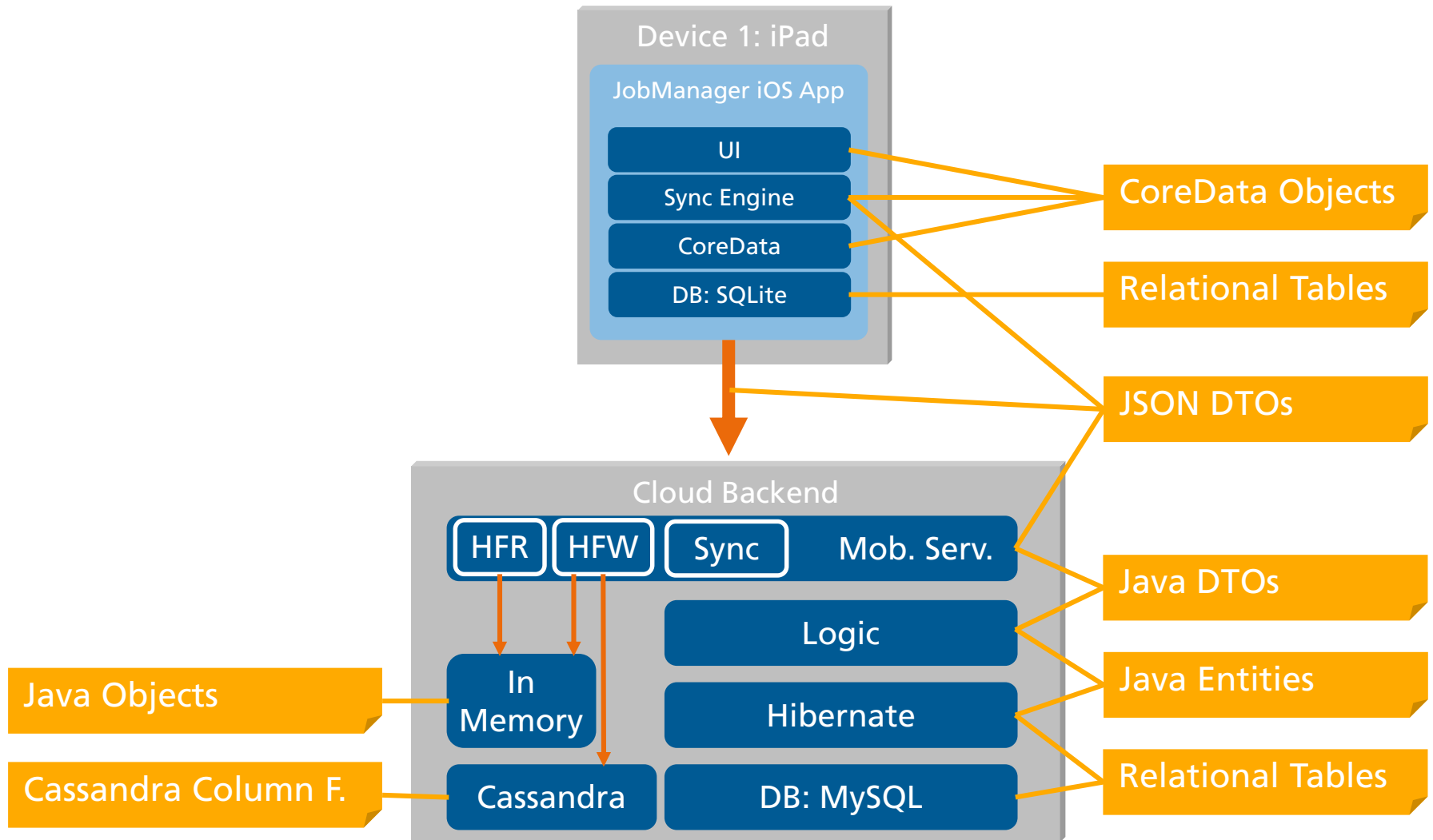
# Multiple Languages to be Supported

12



- Where to transform units?
- In which units to persist?
- Tenant-spanning analyses

# Summary: Overview of Data Representations



# Lessons Learned – Impact on Quality Attributes

**UX and performance:**  
high impact on data architecture

**Security:**  
makes offline capability even more difficult

**Maintainability:**  
nearly always adversely impacted by concepts  
for other quality attributes

# Lessons Learned – Technical Aspects

**Offline capability:**  
not out of the box

**Offline capability:**  
costly to develop

**CoreData:**  
requires in-depth technical know-how

# Lessons Learned – Data Modeling

There is not THE single data model

Data modeling is highly intertwined with architecture design

Data modeling according to OOD / DDD is not enough



# Key Takeaways

**Data is often neglected in architecture**

**Data is highly related to quality attributes**

**Design and document decisions around data!**

**Farming is innovative and strongly uses IT 😊**

architecture

systems

development

software

requirements

product

support

key

competencies

engineering

new

help

documentation

system

terms

order

provide

compliance

services

decisions

design

like

business

generation

projects

evolution

processes

making

practices

Fraunhofer

particular

Tekla

challenges

aspects

basis

architectural

quality

ISE

make

clear

technologies

level

results

many

technology

Systems

organizations

architecture-centric

SAVE

control

evaluation

time

implementation

organization

context

activities

packages

goals

usage

need

one

use

customers

information

models

architectures



Dr. Matthias Naab  
Phone: +49 631 6800 2249  
[matthias.naab@iese.fraunhofer.de](mailto:matthias.naab@iese.fraunhofer.de)



Dr. Ralf Carbon  
Phone: +49 631 3619 1813  
[carbonralf@johndeere.com](mailto:carbonralf@johndeere.com)

# Appendix: Legend of Diagrams

