



CU! - Continuous Usability mit offline-fähigen Apps

Susanne Braun
Matthias Naab

08.03.2016
JavaLand 2016

Fraunhofer IESE

The research institution for software and systems engineering methods

- Founded in 1996, headquartered in Kaiserslautern
- approx. 240 employees
- Our solutions can be scaled flexibly and are suitable for companies of any size
- Our most important business areas:



- Automotive and Transportation Systems
- Automation and Plant Engineering
- Health Care

- Information Systems
- Energy Management
- E-Government

About ACES

ACES –
Architecture-Centric Engineering Solutions

The Fraunhofer Approach for
Modeling Software and System Architectures

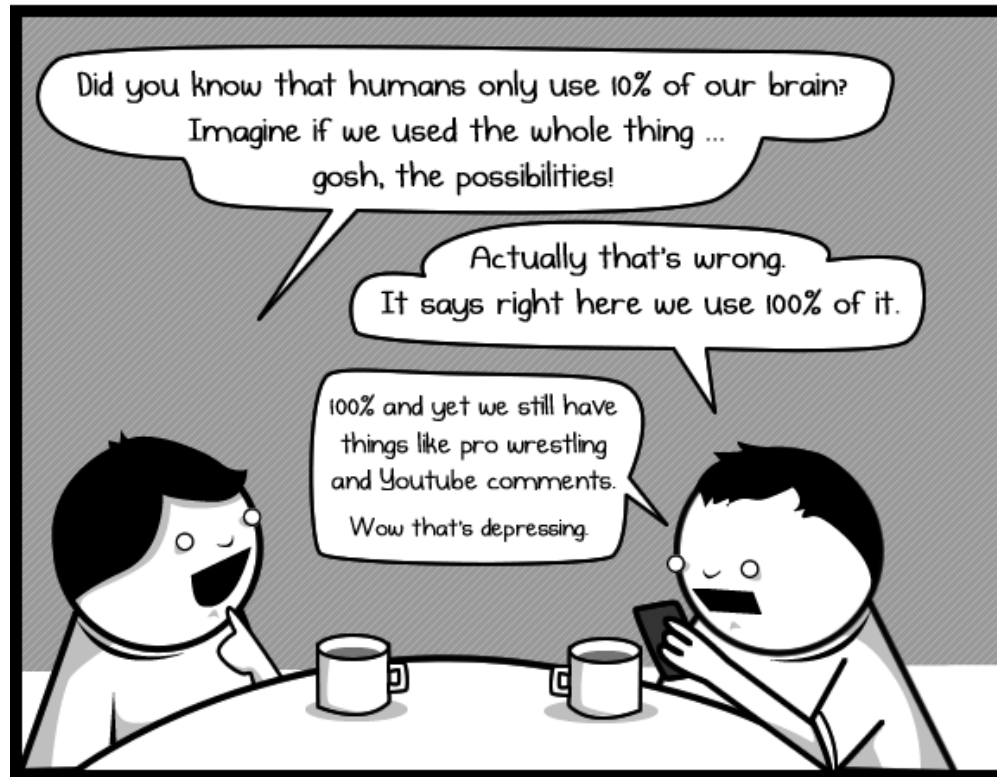
Compiled Best Practices from literature,
scaled and tailored for effective architecting in practice

More than 20 years of architecting experiences across domains:
Embedded Systems, Information Systems, Smart Ecosystems



Always Connected

I love being able to settle any debate



Quelle: <http://theoatmeal.com/comics/smartphone>



Official Blog

Insights from Googlers into our products, technology, and the Google culture

Navigate and search the real world ... online or off

November 10, 2015

Roughly 60 percent of the world is without Internet today, and even where online access is available, it can still be spotty. That means that quick and easy access to information is still not possible for a majority of the population. This is a huge problem, especially as people attempt to navigate and explore the world around them, so Google Maps is taking steps to help people across the globe find directions and get where they're going, even when they don't have an Internet connection.

User Experience Expectations of Mobile Users

And I hate listening to people complain about their phones.



If you really think your smartphone is a piece of junk, by all means switch back to a Motorola RazR. →

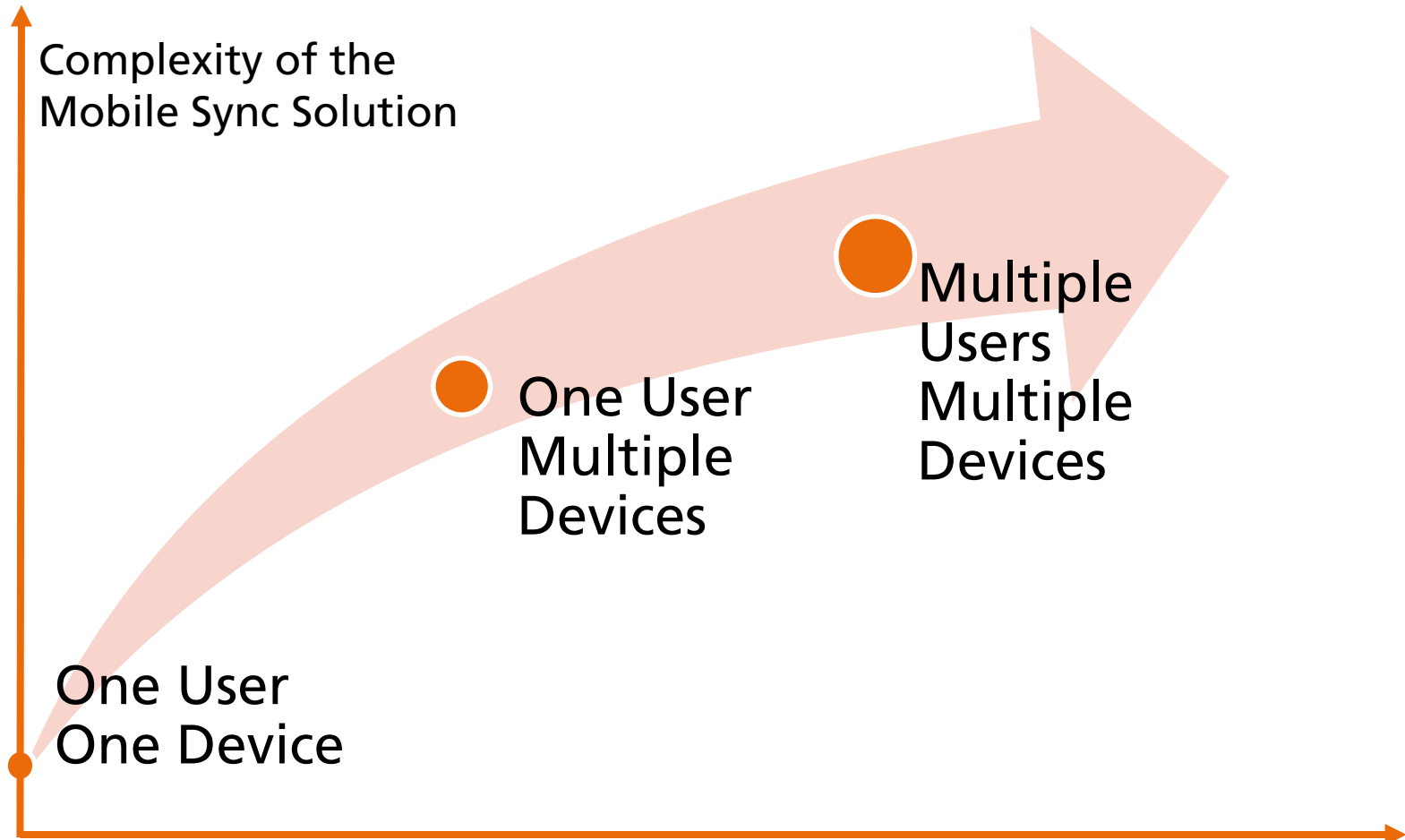


What Mobile Users expect from high quality Apps

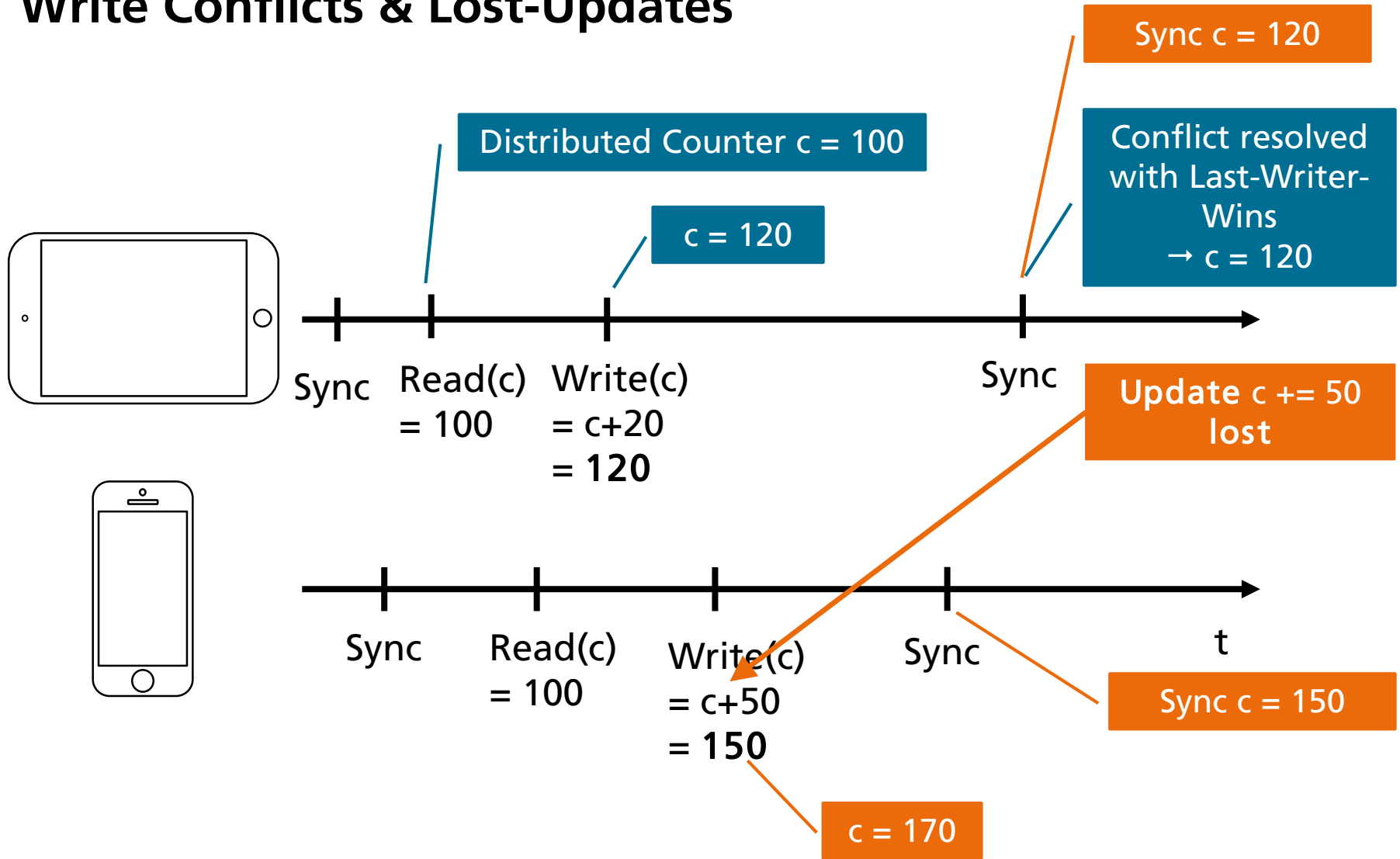
- Continuous Availability of Mobile Services and Data
- Responsive and Polished UIs
- Economic Consumption of Battery & Data Plans

**Apps need to be offline usable
and ship with smart & economic
data synchronization**

Mobile Data Synchronization Classes



Write Conflicts & Lost-Updates



ACID & Replication

The Dangers of Replication and a Solution

Jim Gray (Gray@Microsoft.com)
Pat Helland (PHelland@Microsoft.com)
Patrick O'Neil (POneil@cs.UMB.edu)
Dennis Shasha (Shasha@cs.NYU.edu)

Abstract: *Update anywhere-anytime-anyway transactional replication has unstable behavior as the workload scales up: a ten-fold increase in nodes and traffic gives a thousand fold increase in deadlocks or reconciliations. Master copy*

Eager replication delays or aborts an uncommitted transaction if committing it would violate serialization. Lazy replication has a more difficult task because some replica updates have already been committed when the

„Update anywhere-anytime-anyway **transactional replication** has **unstable** behavior as the workload scales up: a ten-fold increase in nodes and traffic gives a **thousand fold** increase in deadlocks or reconciliations.“

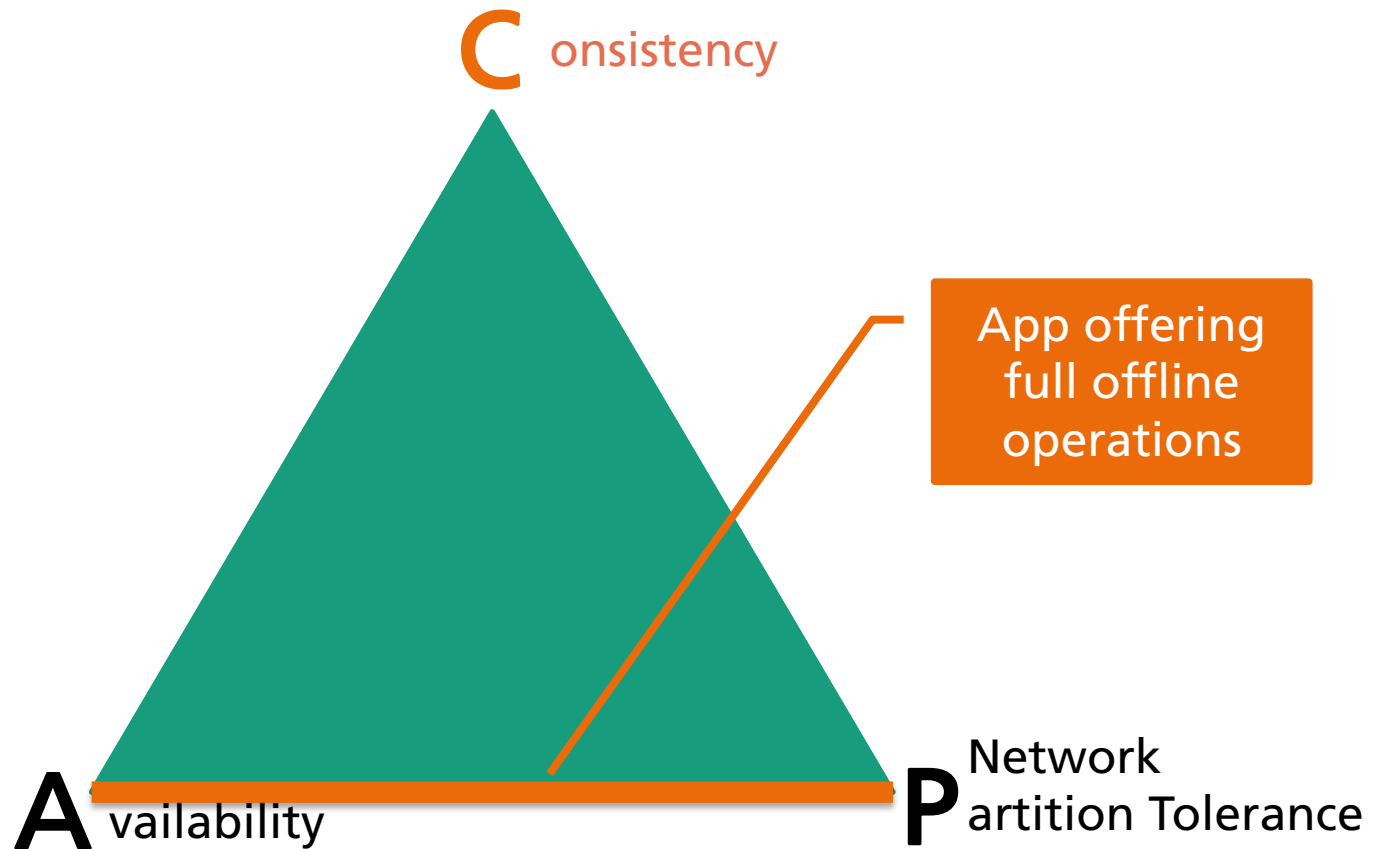
Jim Gray, 1996*

transaction response times because extra updates and messages are added to the transaction.

checks totaling \$1,000 for a total of \$2,000 in withdrawals. When these checks arrived at the bank, or when you communicated with your spouse, someone or something

* <http://research.microsoft.com/apps/pubs/default.aspx?id=68247>

CAP Theorem*



* Eric Brewer

How to achieve Convergence?

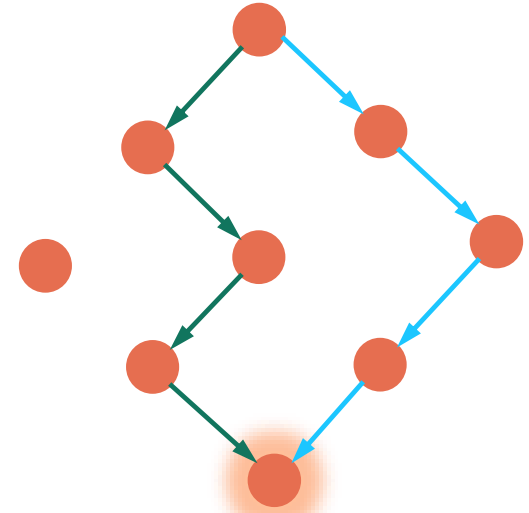
Anti-Entropy

- Exchange data versions or update operations between replicas

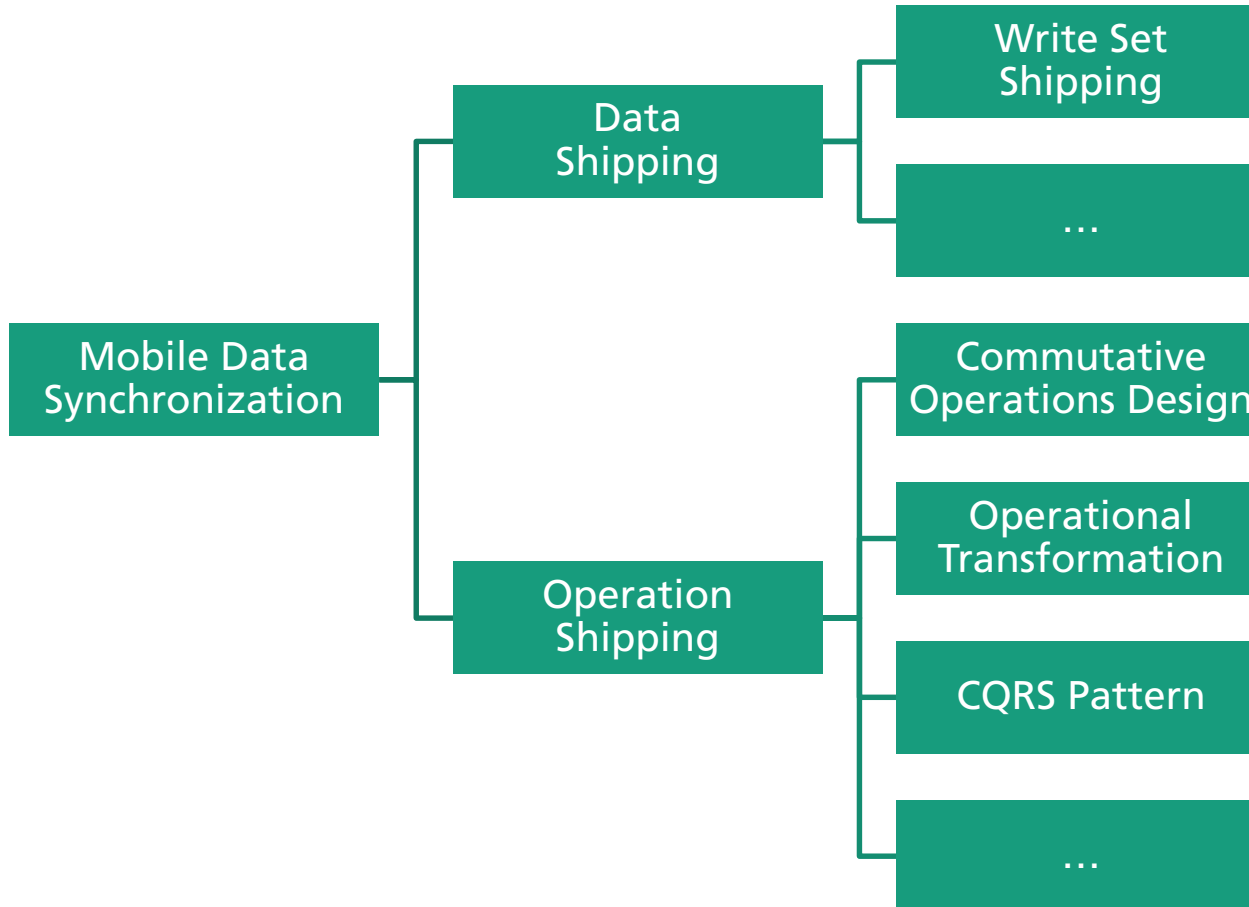
Reconciliation

- Conflict Detection
- Conflict Resolution

Highly
Application
Dependent

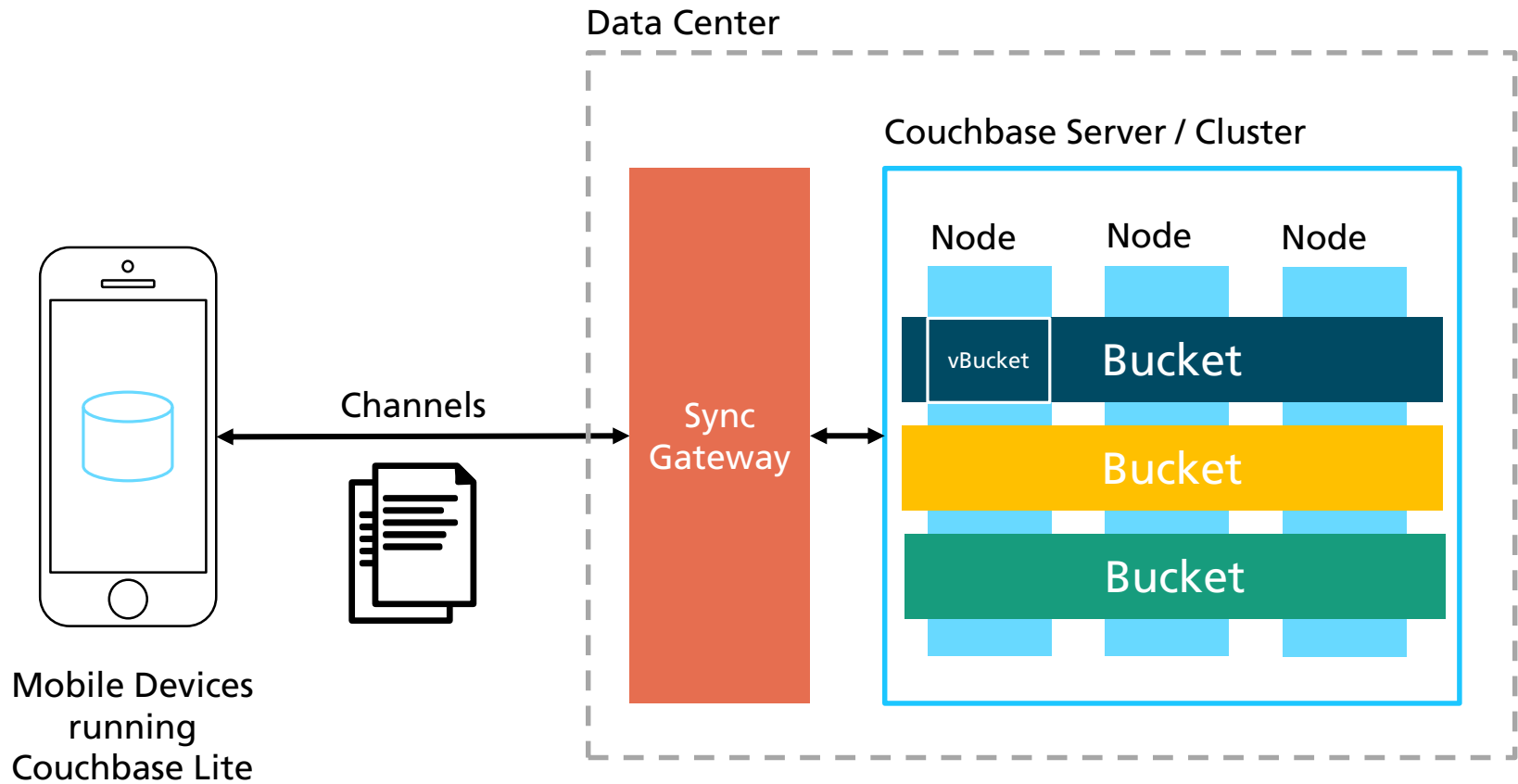


Approaching Mobile Data Synchronization

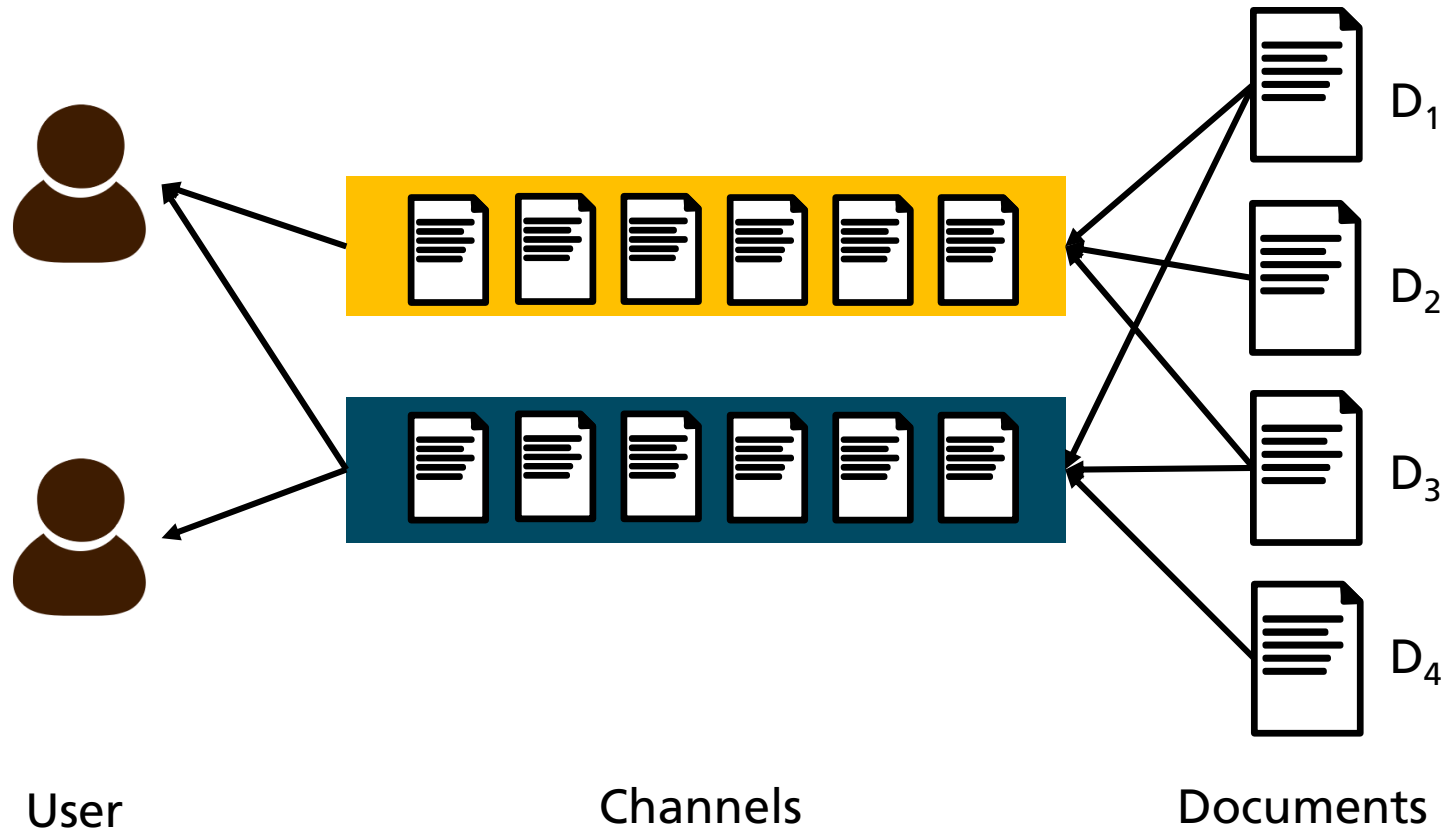


Couchbase Mobile

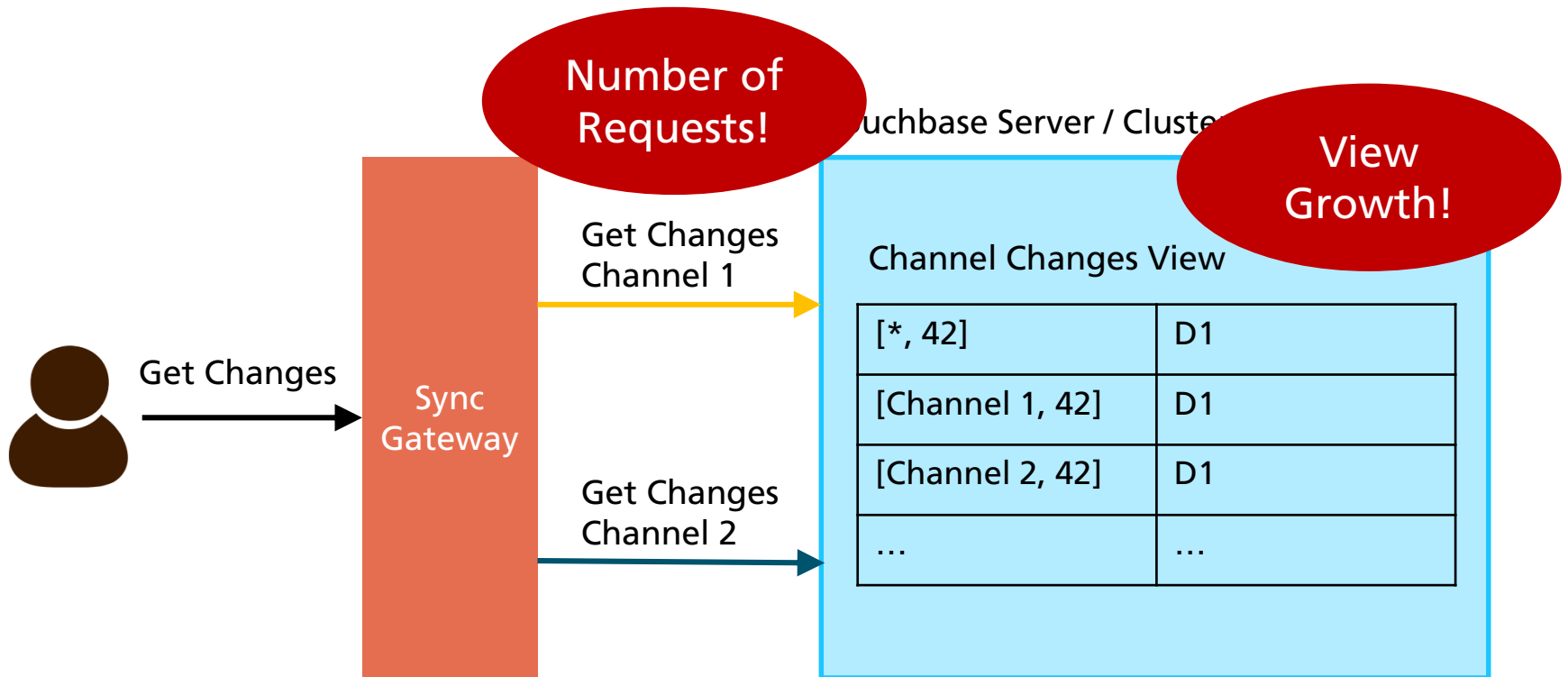
Couchbase Mobile



Couchbase Mobile Channels

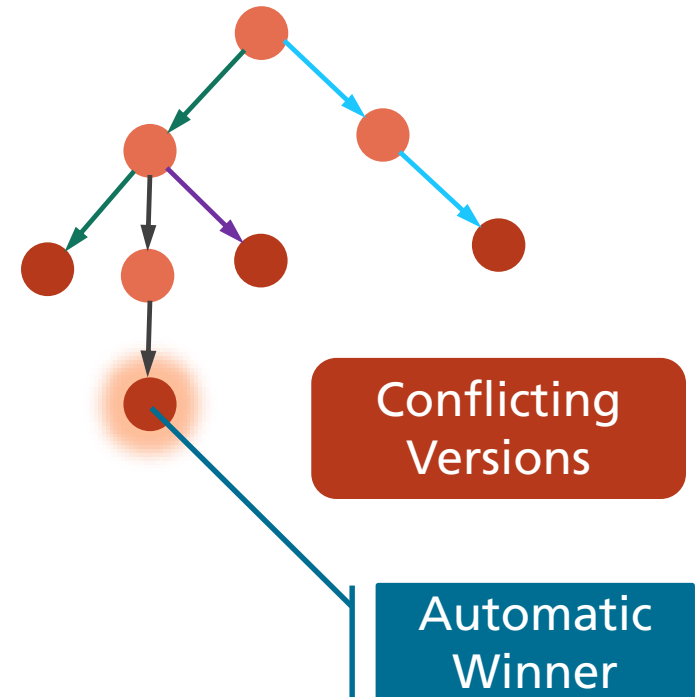


Couchbase Mobile Scalability



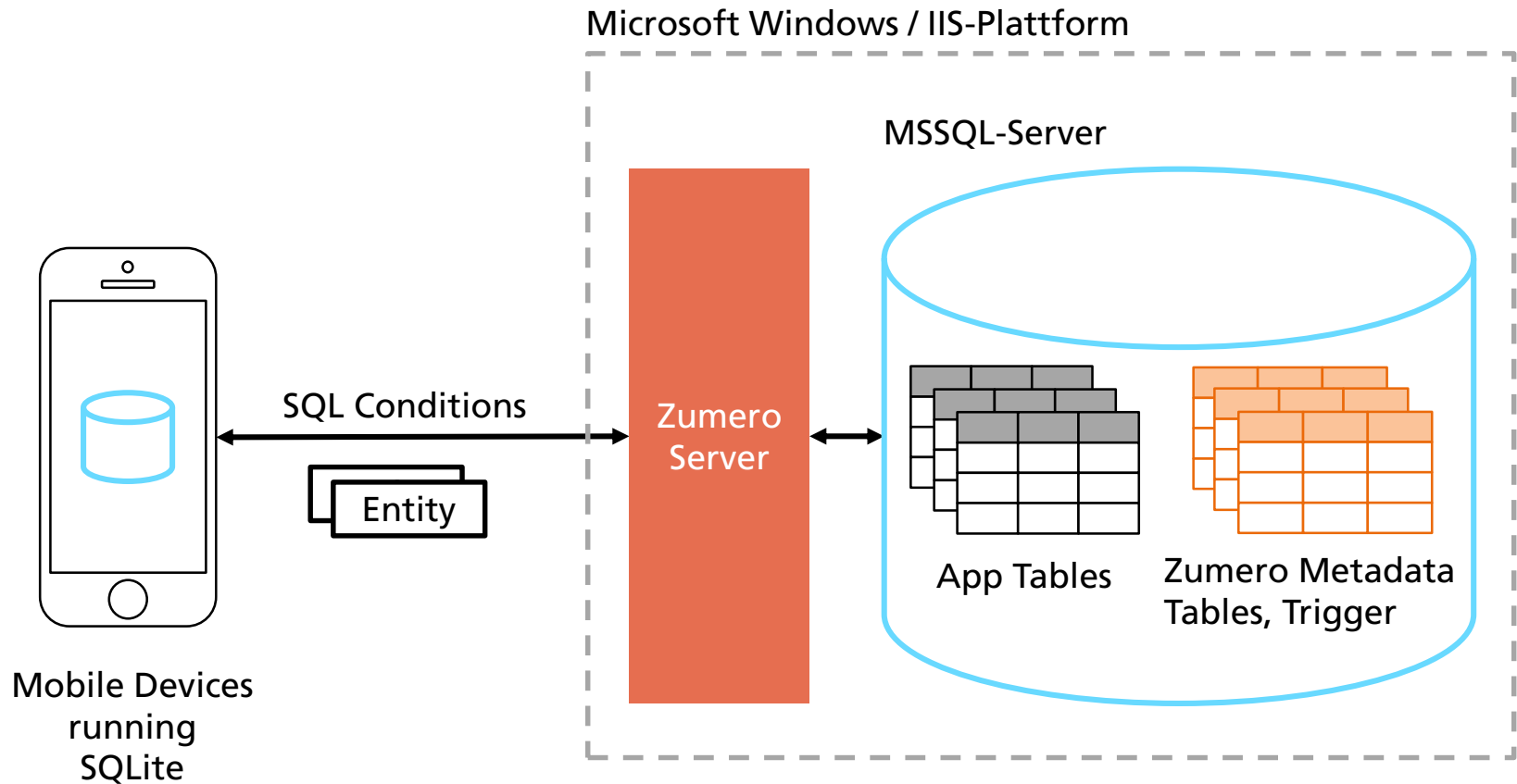
Couchbase Mobile Conflict Resolution

- Automatic and deterministic determination of winner version
- Without communication overhead
- Winner:
 - Undeleted version with greatest depth in version graph
 - If more than one: lexicographic comparison on revision
 - „Busiest Writer“ wins or arbitrary in worst case
- Support for manual conflict resolution
- Guarantee that conflicting versions are available
- No Guarantee that all historic versions are available



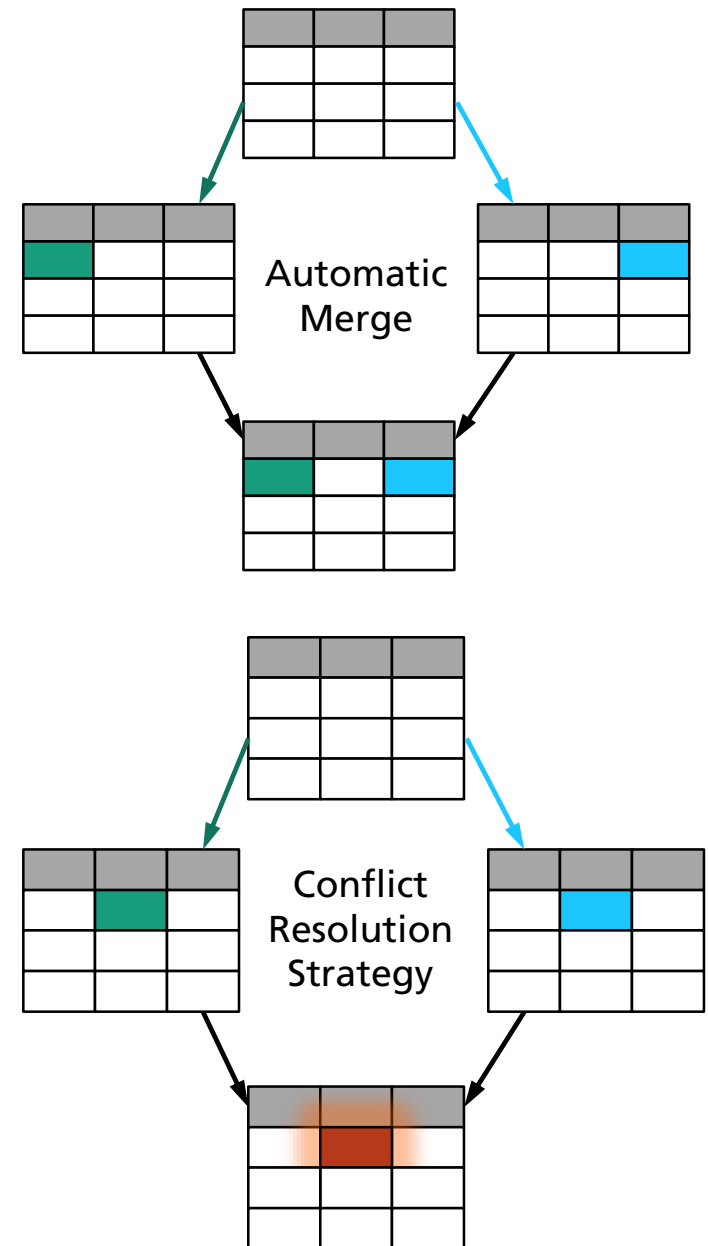
Zumero

Zumero



Zumero Conflict Resolution

- Support for **automatic merge** of updates on different entity column values
- Database Constraint Violations cannot be automatically merged
- Strategies for conflicting updates on same entity column:
 - Last-Writer-Wins
 - First-Writer-Wins
- Support for manual conflict resolution
- Guarantee that conflicting entities are available
- Historic entity versions are maintained in metadata tables (unofficial & undocumented)



Products & Frameworks

Frameworks



Cognito



Android Sync Adapter
Framework

One User,
Multiple Devices

DB Replication



Couchbase



Zuora

Multiple Users,
Multiple Devices

A person wearing blue denim jeans and brown leather boots is standing in a field of tall grass. The background shows a sunset over a hazy landscape with trees and hills. The sun is low on the horizon, creating a warm, golden glow. The text "My John Deere Mobile Case Study" is overlaid in white on the person's legs.

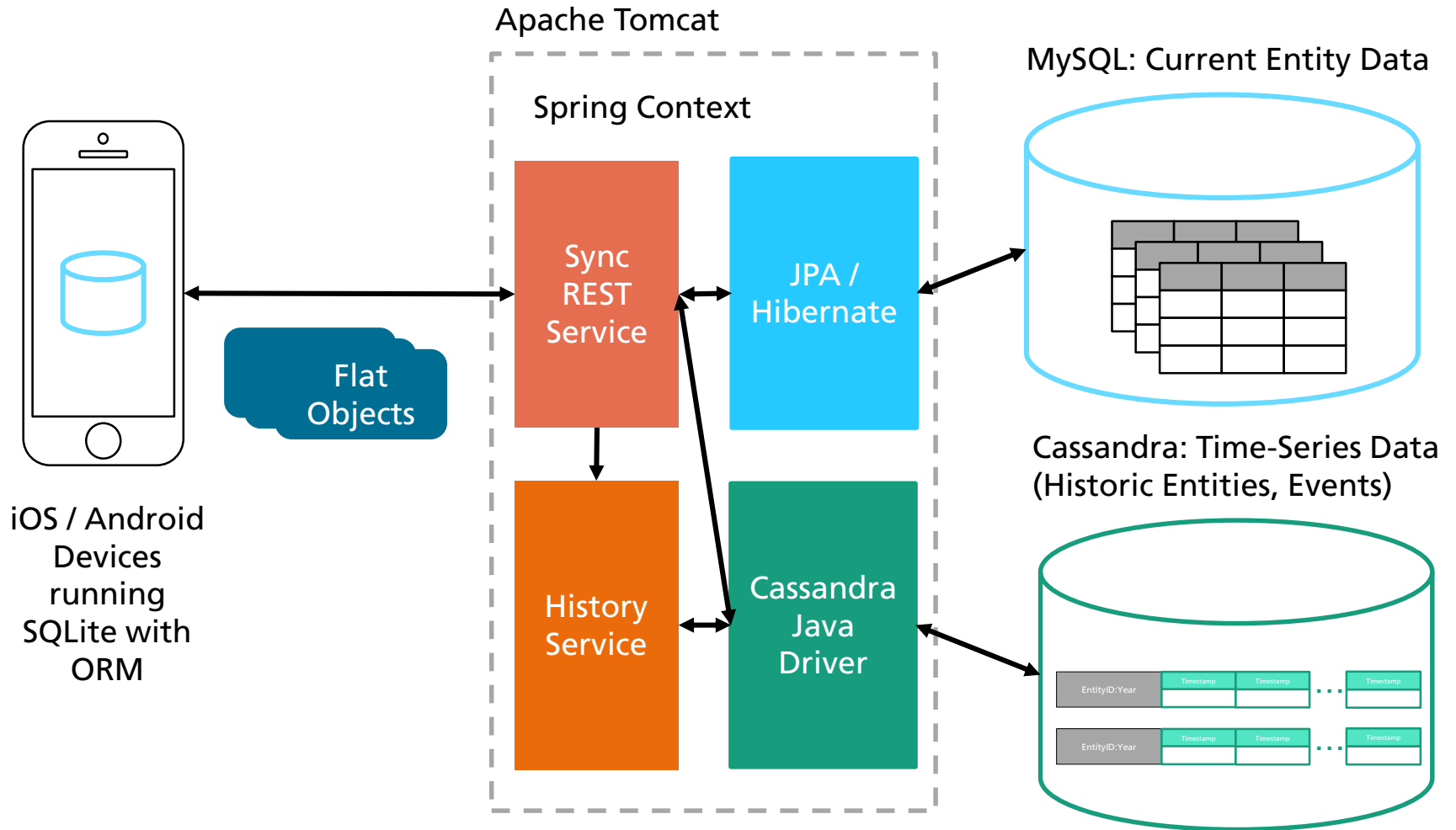
My John Deere Mobile Case Study

Important Design Decisions

- **Consistency Guarantees** (What are the desired Consistency Guarantees? How are they achieved?)*
- **Update Format** (Exchange of data items vs. exchange of update operations?)*
- **Change Tracking** (How to track updates that need to be propagated?)*
- **Metadata** (What metadata is stored and communicated about replicated items?)*
- **Sync State** (What state is maintained at a device for each synchronization partnership?)*
- **Change Enumeration** (How do devices determine which updates still need to be propagated to which other devices?)*
- **Communication** (What transport protocols are used?)*
- **Ordering** (How to decide on the order in which received updates should be applied?)*
- **Filtering** (How are the contents of partial replicas specified and managed?)*
- **Conflicts** (How are conflicting updates detected and resolved?)*
- **Synchronization Time** (How to decide on the optimal point in time for synchronization? Impact on battery consumption, data traffic and probability for conflicts.)

* Source: Replicated Data Management for Mobile Computing, Douglas Terry, p. 27

MJDM Sync Solution Architecture



Change Tracking & Change Enumeration

JPA Entity Base Class

- id: UUID
- syncState: updated | inserted | deleted | insync
- revNr: numeric version

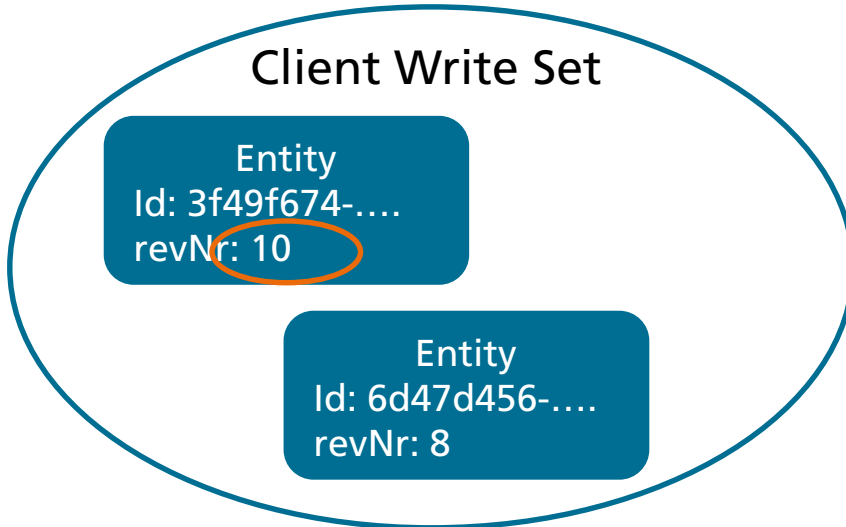
Globally unique identifier

- Version number of the entity
- Managed by Sync Service
- Enumerate changes on the server
- Conflict Detection

- Track and enumerate changes on the client
- Mark entities as deleted on client and server

Conflict Detection

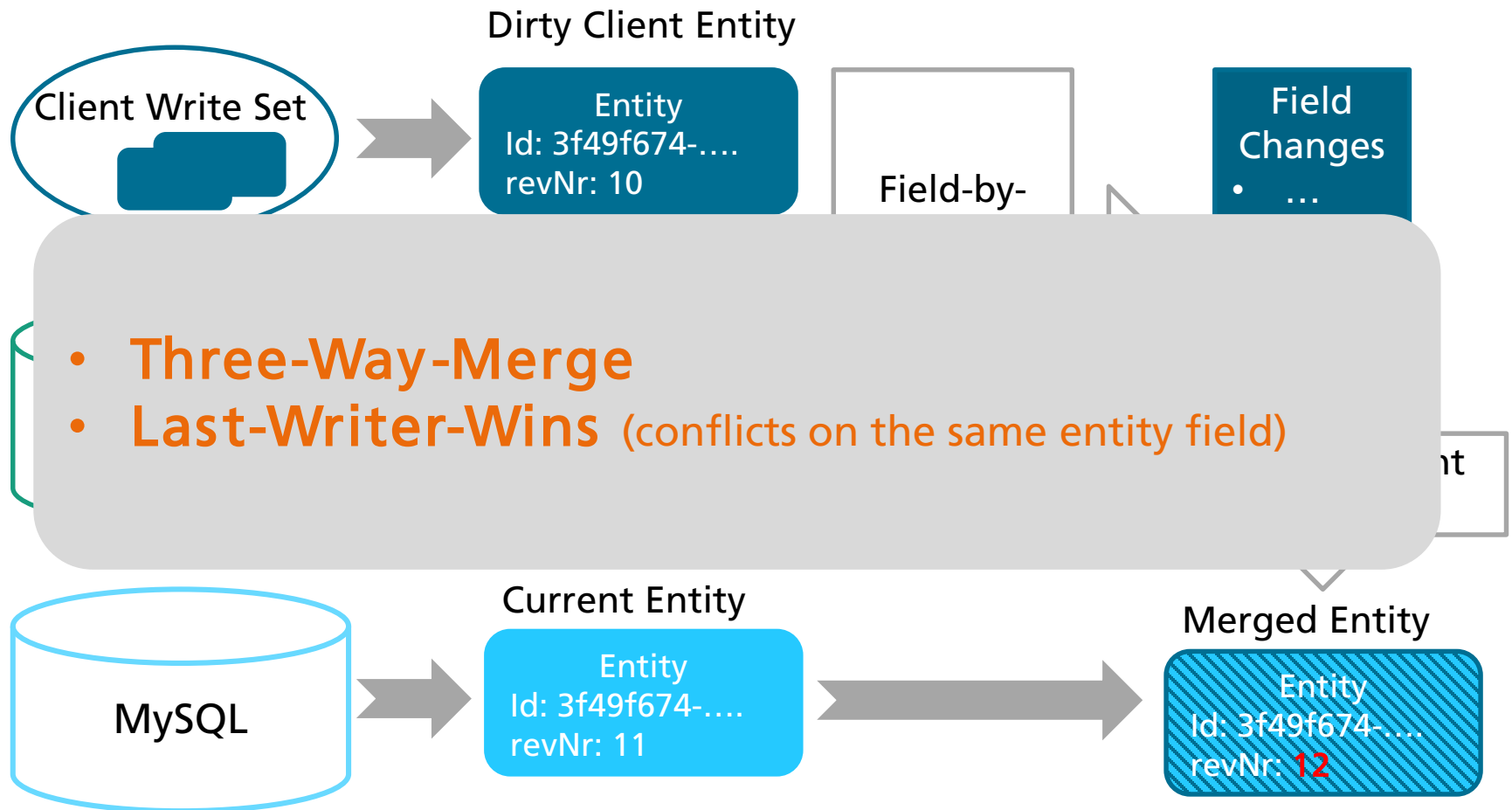
- Client must keep and ship revNr of each entity in the Write Set
- Server preprocess Write Set of Client
- Server iterates over all entities in Write Set and
- Checks if **revNr of Client < revNr of Server**
- → **Conflict**



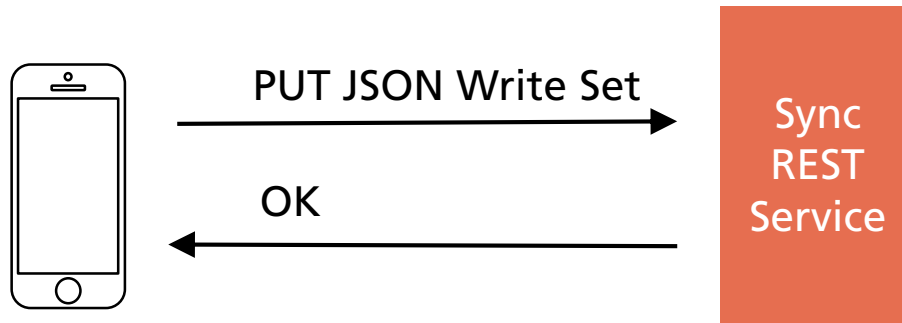
Server MySQL DB

ID	REV	...
3f49f674-....	11	
6d47d45-....	8	

Conflict Resolution

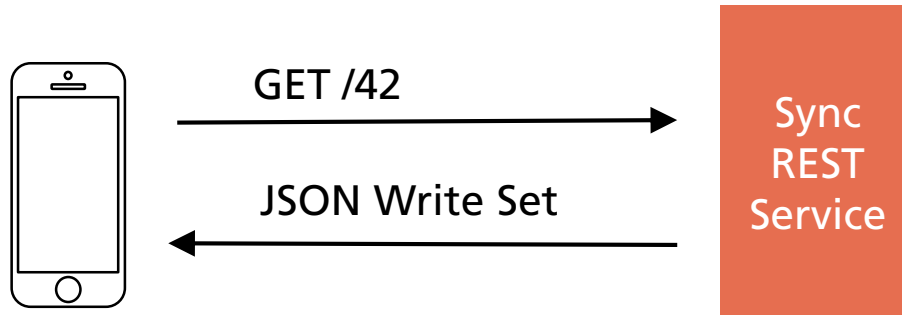


Two-Phase Protocol



Atomic

- Conflict detection
- Conflict resolution
- Increase sync counter
- Update revision number of entities
- Update syncState (insync | deleted)
- Persist entities
- Insert historic entity versions



Atomic

- Calculate Write Set
- Entities with revision number > 42
- Entities either deleted or insync

- All entities „insync“
- syncState deleted entites are now deleted on client

Caution: Avoid Disadvantageous Data Modelling



- Insertion of Ag Plan
- = Collection update in Field
- → New version of Field
- → **Can conflict** with other Field updates.

- Insertion of Ag Plan
- = Insertion of Ag Plan!
- -> **Cannot conflict** with any other update!

Lessons Learned

Impact on Data Modelling

Impact on Code Complexity

The Return of the Fat Clients
in conjunction with Data-Centric
Architectures...

Data Validation can become a Show
Stopper

Write Set Shipping Cries for Generative
Approaches

Authorization Changes Usually Require a
Full Sync

Full Sync Usually Requires Special
Optimization

Historic Data is Valuable

Design for Economic Resource Consumption

Design for Great User Experience
is not trivial...

“By the end of 2014, the number of mobile-connected devices will exceed the number of people on earth”¹

“Last year’s mobile data traffic was nearly 30 times the size of the entire global Internet in 2000”¹

“The Future of Enterprise Applications Is Mobility”²

Questions?