CumuloNimbo
3rd Year Final Review Meeting
Brussels, Nov 27th, 2013

Overall Overview
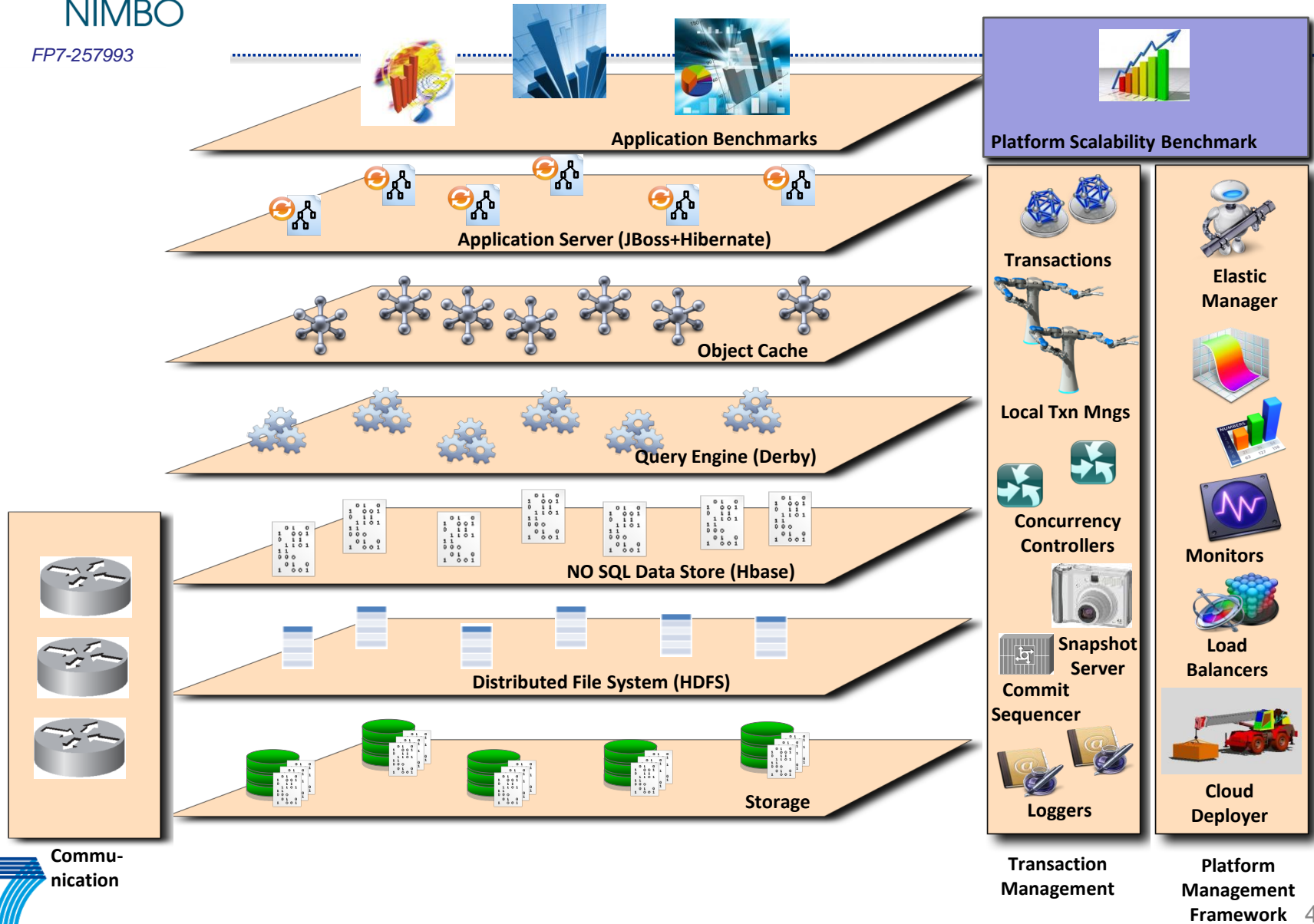
Ricardo Jiménez-Peris
UPM

Motivation

- Current cloud PaaS attains scalability by means of new software stacks (e.g. no SQL data stores) and sharding.

- This results in having to substantially modify applications or rebuild them from scratch.

- Even after rebuilding applications still end users will be suffering for the lack of full transactional semantics due to sharding.

- This is one of the main obstacles for many applications to be moved to or created for the cloud.

Goals

- CumuloNimbo precisely addresses this issue.
- CumuloNimbo aims at architecting and developing a transactional PaaS that:
  - Scales to 1000s of cores.
  - Simple application migration to the cloud by means of transparency.
  - Achieve syntactical transparency:
    - Existing applications can be moved to CumuloNimbo PaaS without modification.
  - Guarantees semantic transparency:
    - The applications observe exactly the exact semantics as in a traditional centralized environment.
  - Provides elasticity and high availability.
  - Does all the above efficiently.
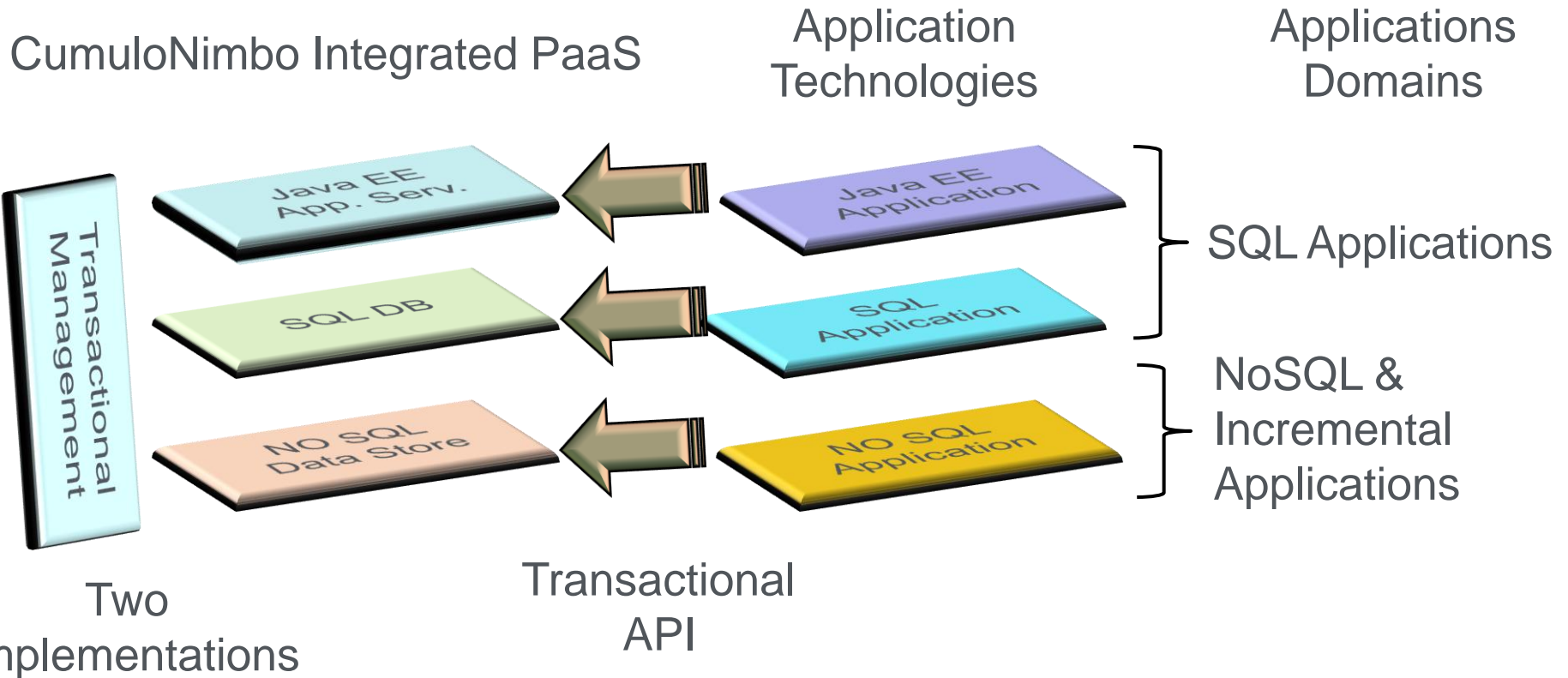
CumuloNimbo Architecture

Application Benchmarks

Platform Scalability Benchmark

Application Server (JBoss+Hibernate)

Object Cache

Query Engine (Derby)

NO SQL Data Store (Hbase)

Distributed File System (HDFS)

Storage

Commu-nication

Transactions

Local Txn Mngs

Concurrency Controllers

Commit Sequencer

Snapshot Server

Loggers

Transaction Management

Elastic Manager

Monitors

Load Balancers

Cloud Deployer

Platform Management Framework

Extended Scope of the Project

- The initial scope of the project only addressed support for a single software stack:

    – Multi-tier applications (e.g. Java EE applications).

- The project has extended its scope to support two additional software stacks:

    – Relational database clients (SQL interface).

    – Big data and incremental application clients (Hbase interface).

- This has significantly increased the impact of the project.

CumuloNimbo:
An Integrated Transactional PaaS

CumuloNimbo Integrated PaaS

Application Technologies

Applications Domains

Transactional Management

Java EE App. Serv.

SQL DB

NO SQL Data Store

Java EE Application

SQL Application

NO SQL Application

SQL Applications

NoSQL & Incremental Applications

Two implementations

Transactional API

Advancement with Respect to SOTA

FP7-257993

- Ultra-scalable transactional processing.
- Syntactical and semantic transparency for the scalability.
- Low intrusive elasticity and dynamic load balancing.
- Low cost fault-tolerance.
- Lower overhead storage.
- Novel architecture:
  - Decomposition of transactional processing.
  - SQL processing on top of No SQL data stores.

Focus of Third Year

- Integration of all partner contributions.
- Building demos exercising all components in the different stacks.
- Using elastic benchmarking to inject the load.
- Deployment, monitoring and elastic management via PMF.
- Exercising Flexiant FCO at UPM with local storage management.
- Exercising Flexiant public cloud.

Achievements

- Original goals exceed by adding new contributions.

- Prototypes for all contributions ready.

- Full integration achieved.

- Demos running on FCO@UPM and Flexiant public cloud.

- All developed components used and demonstrated.

- Numerous improvements made during the last year as well to improve the efficiency, quality and functionality of the prototypes.

- A startup will be created to commercialize some of the main innovations from the project in which several partners will participate.