CORAL Server

A middle tier for accessing relational database servers from CORAL applications

A. Valassi, A. Kalkhof, D. Duellmann, Z. Molnar (CERN IT-DM)

M. Wache (University of Mainz / Atlas)

A. Salnikov, R. Bartoldus (SLAC / Atlas)

CHEP2009 (Prague), 23rd March 2009

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Outline

- **Motivation**
- **Development status**
  - Software architecture design
- **Conclusions**

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Introduction

- **CORAL used to access main physics databases**
  - Both directly and via COOL/POOL
    - Previous talk about LCG Persistency Framework (A.V.)
    - Important example: conditions data of Atlas, LHCb, CMS
  - Oracle is the main deployment technology at T0 and T1
    - Previous talk about distributed database operation (M. Girone)
- **No middle tier in current client connection model**
  - Simple client/server architecture
- **Limitations of present deployment model**
  - Security, performance, software distribution (see next slide)
  - *Several issues may be addressed by adding a middle tier*

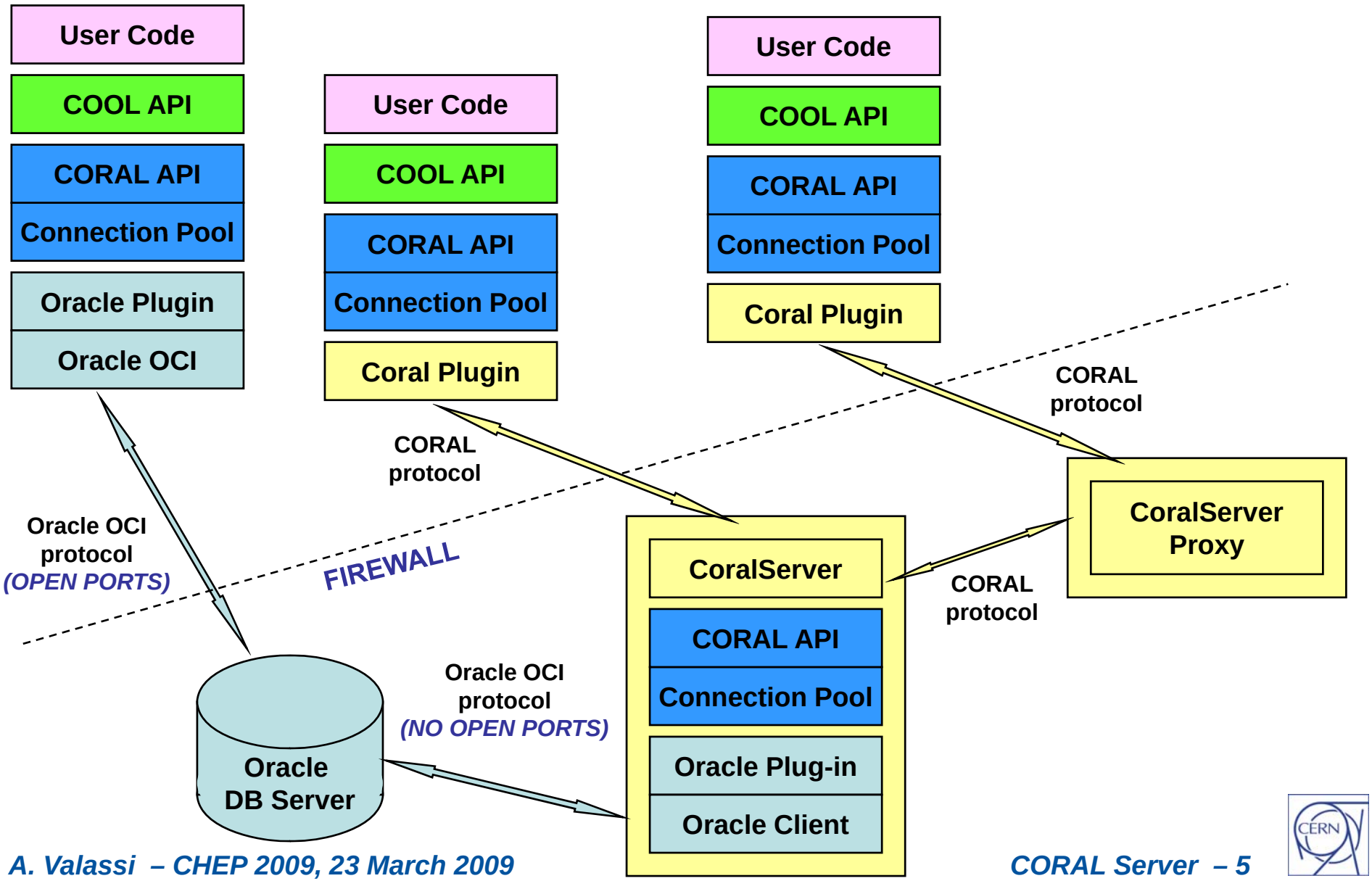
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Motivation

- **Secure database access**
  - Authentication via Grid certificates
    - No support of database vendor for X.509 proxy certificates
    - Hide database ports within the firewall (reduce vulnerabilities)
  - Authorization via VOMS groups in Grid certificates
- **Efficient and scalable use of server resources**
  - Multiplex client connections using fewer connections to DB
  - Option to use additional caching tier (CORAL server proxy)
    - Also useful for further multiplexing and for client monitoring
- **Client software deployment**
  - CoralAccess client plugin using custom network protocol
    - No need for Oracle/MySQL/SQLite client installation
- **Interest from several stakeholders**
  - Service managers: physics DB team, security team...
  - LHC users: Atlas HLT (replace MySQL-based DbProxy)...

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Deployment scenario (e.g. COOL)



Development overview

CERN IT Department

- **First phase of development in 2007-2008**
  - User requirements (Atlas HLT) analyzed in Nov-Dec 2007
  - Main emphasis of development on server components
    - Design based on template meta-programming techniques
  - Success in COOL R/O tests, problems in Atlas HLT tests
- **New developments in 2009** ← ***Main focus of this talk***
  - Several changes in the team in December 2008
    - Software review to identify areas for improvement
    - Development restarted in Jan 2009 using a new comprehensive architecture design for server and client
  - Keep current focus on R/O access with proxy for Atlas HLT
    - This is only the first customer – but is an excellent benchmark
  - Promising outlook after only two months
    - Success in multi-client HLT tests against COOL Oracle DB

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

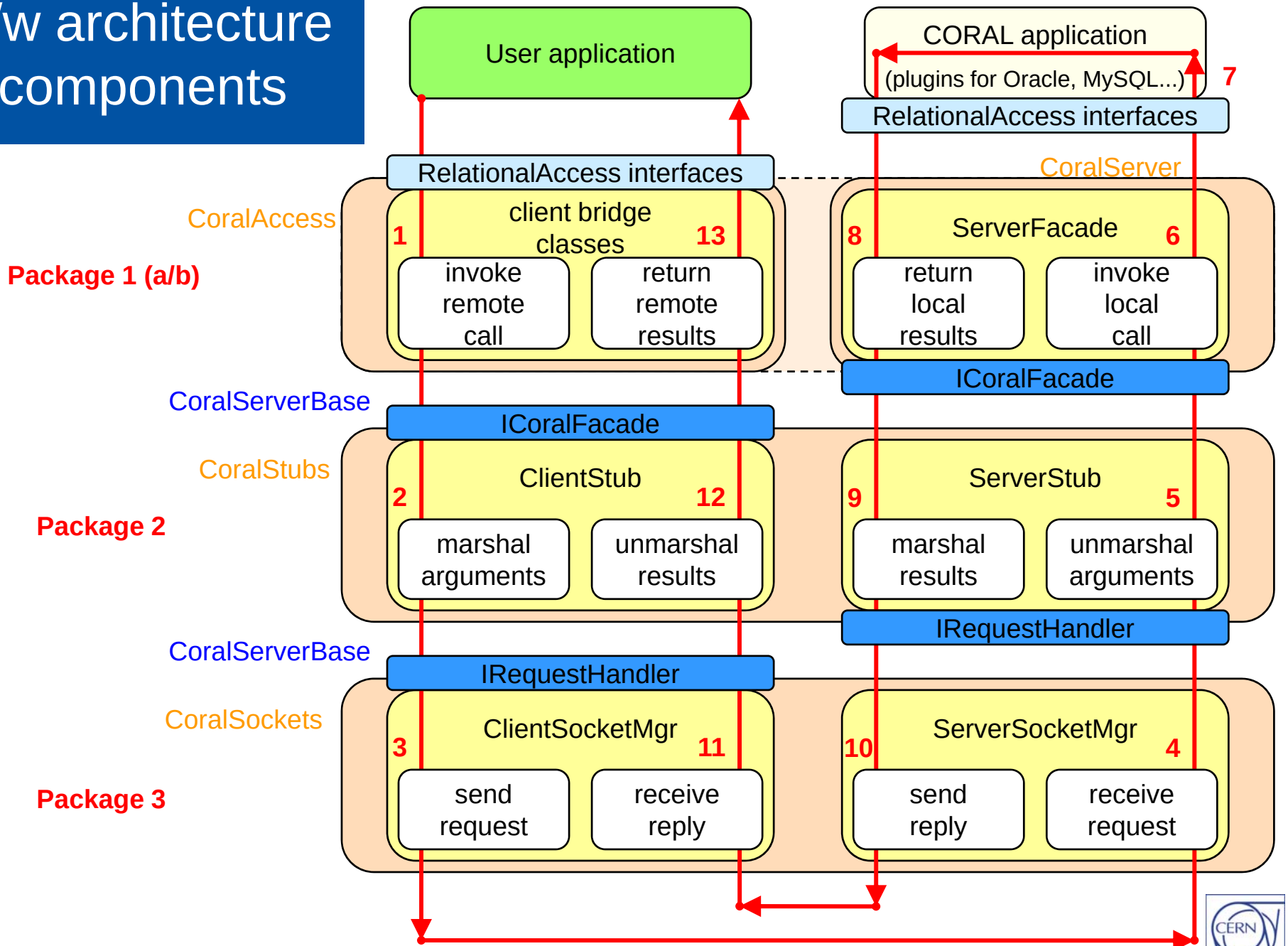
Guidelines for the new design

CERN IT Department

- **Joint design of server and client components**
  - Split system into packages ‘horizontally’ (each package includes a pair of one server and one client components)
  - RPC architecture based on Dec2007 python/C++ prototype
- **Allow several people to work in parallel**
  - Minimize software dependencies and couplings
  - Upgrades in one package should not impact the others
- **Include standalone package tests in the design**
  - Aim to intercept issues before they show up in system tests
- ***Decouple components using abstract interface***
  - Modular architecture based on object-oriented design
  - Thin base package with common abstract interfaces
  - Encapsulate implementation details of concrete packages

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

S/w architecture components



A few implementation details

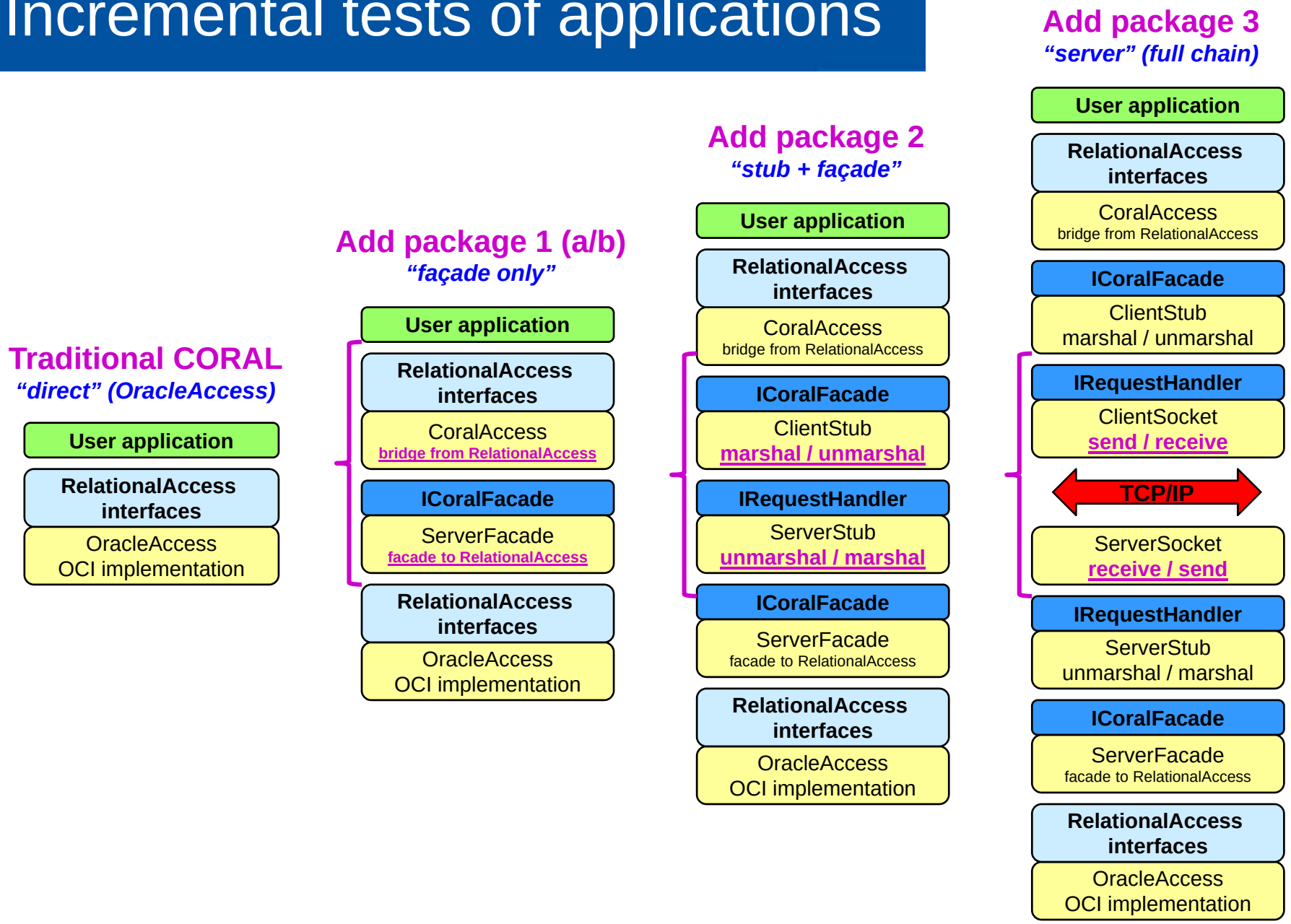
CERN IT Department

- **Multi-threaded server**
  - Threads are managed by the SocketServer component
  - One listener thread (to accept new client connections)
  - One socket thread per client connection
    - Pool of handler threads (many per client connection if needed)

- **Network protocol agreed with proxy developers**
  - Most application-level content is opaque to the proxy
    - Proxy understands transport-level metadata and a few special application-level messages (connect, start transaction…)
    - Most requests are flagged as cacheable: only hit the DB once
  - Server may identify (via a packet flag) client connections from a proxy and establishes a special ‘stateless’ mode
    - Single R/O transaction spans the whole session
    - ‘Push all rows’ model for queries (no open cursors)

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Incremental tests of applications



Current development status

CERN IT Department

- **Focus on R/O access with proxy for Atlas HLT**
  - Tests with Atlas HLT COOL db ***(OK)***
  - Tests with Atlas HLT geometry db *(ongoing)*
  - Tests with Atlas HLT trigger db *(in preparation)*
  - Atlas HLT data validation *(not started yet)*
  - Atlas HLT performance validation *(ongoing)*
- **Observed hangs with Oracle connection sharing**
  - Libraries are not thread-safe (CORAL, Oracle client)?
  - Connection sharing may be disabled in Atlas HLT
    - But it is essential for connection multiplexing without a proxy
- **Progress on secure authentication/authorization**
  - OpenSSL with VOMS handling of proxy certificates

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Outlook

- **First release (R/O, no certificates) in Q2 2009**
  - After we have fulfilled the Atlas HLT requirements
- **Support for proxy certificates in Q2/Q3 2009**
  - Secure authentication and authorization
- **Full R/W functionalities in Q3 2009**
  - DML (e.g. insert table rows) and DDL (e.g. create tables)
- **Deployment of test servers at CERN by Q4 2009**
  - In parallel, discuss deployment at T1 sites

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Reserve slides

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Atlas HLT and DbProxy

CERN IT Department

DbProxy in a Nutshell

- **Original Scope: ATLAS High Level Trigger (HLT) Configuration**
  - Must configure entire farm of L2+EF processors within O(1-10) s
  - O(2000) HLT nodes × 8 cores/node × 1 client/core; O(10-100) MB of configuration data/client → O(0.1-1) TB of data being read
  - Must reduce number of connections (→ multiplexing) and network traffic (→ caching)
- **Characteristics**
  - Process that sits between HLT client and database; acts as a server downstream and as a client upstream
  - Transparent to the HLT client that connects to it
  - Can itself connect to another proxy; *i.e.*, allows one to build hierarchies (tree of proxies)

Rainer Bartoldus *DbProxy Status*, Online Database Meeting, Apr 21, 2008

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Architecture (client/server)