



TripCom

WP6 – Team and Work Package Presentation

Vienna, 24.04.2006





The Research Group &



Content of Work Package 6

- SBC Research Group at TUW:

Leader: eva Kühn

PhD Students: Marcus Mor, Richard Mordinyi,
Martin Murth, Johannes Riemer, Fabian Schmied

Senior Researcher: Geri Joskowicz

- Recap of WP 6 Content:

- (1) Collaboration procedures
- (2) Architecture / Scope and responsibilities of components
- (3) Middleware / Extension of Space Technology
- (4) Reference Architecture & APIs
- (5) Integration
- (6) Evaluations in respect to Use Cases

The required activities focus around 3 categories:

■ **Software Development Methodology & Design**

- (1) Collaboration Procedures
- (2) Architecture & scoping of components
- (3) Reference Architecture & API's
- (4) Evaluation in respect to Use Cases (Component level)

■ **Design, Implementation & Delivery of a component**

- (1) Middleware /Extension of Space Technology

■ **System Integration**

- (1) Integration of a complete system
- (2) Evaluation in respect to Use Cases (Overall System level)

Collaborative Software Development / Methodology

■ Challenges:

- **Many stakeholders with individual contributions**
Contributions are based on complete systems; Large number of components, some with strong architecture/framework definitions
- **Definition of In's & Out's:**
Identification of overlaps; scoping of components; component boundaries, ...
- **Collaborative specification & design methodologies:**
Adoption of Specification standards;
Iterative introduction of industrial grade development processes; ...
- **Complex work package dependencies:**
Constraints on schedule of each group; Avoidance of critical dependencies

■ Practical and agreed solutions within 6 Months !!!

Development Methodology Solution Principles (Proposal)

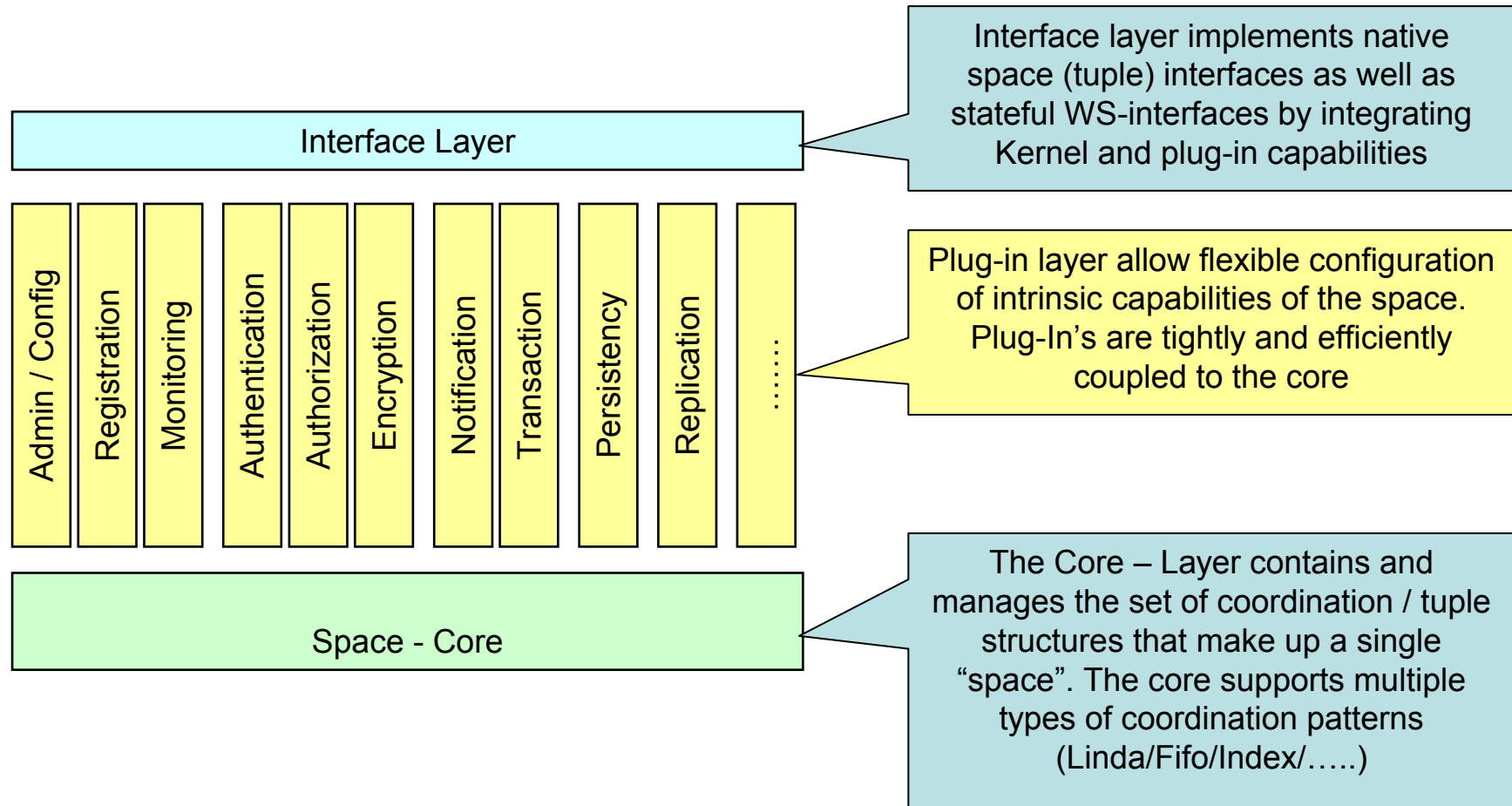
- Collaborative, Iterative Process:
 - There is no silver bullet –
Methodologies are acquired in steps and validated through practice.
 - Derivation from existing development processes;
Understanding of current working practices in contributing groups
 - Continuous improvement and extension as we go along.
Avoidance of unnecessary formalisms; Applicability is first priority
- Kick-starting the process definition through “seeding”:
 - “Seed:” Use of the component to be delivered by WP6 as a working example, validation and test bed for the definition of the collaborative process.
- Architecture coordination:
 - Requires each contributor to name a responsible architect
 - Intensive and frequent coordination (workshops, tele-conferences, ..)
 - Creation of a repository for software artefacts: Administration, Versioning, ...



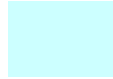
Design/Implementation of Space based Infrastructure



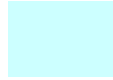
- Contribution of TUW's research group is based on CORSO – a coordination system that evolved over the last 10 years.
- Proven in real world applications
- New, redesigned architecture to provide cleaner layering, higher flexibility (plug-in's) and simpler interfacing.



- Adaptation to TripCom (Infrastructure View)
 - Open functional issues:
 - Additional Plug-In's or repackaging of contributed components as Plug-In's
 - Interface definitions and integration features
 - Open non-functional issues:
 - Scalability
 - Interface standards (Stateful WS standards, other standards)



- Challenges:
 - “Local acting – global thinking”, e.g. maximum freedom / ownership for each contributing group while preserving the common goal of a powerful, integrated system.
 - Integration of complex systems are notoriously the weakest link in multi-site projects (- many examples)
 - Allocation of classic integration features like:
 - Configuration layer & management
 - Common management layer



- Proposed solution principles
 - Incremental approach as in development methodology, iterative process starting with simple integrations.
 - “Use cases \leftrightarrow Integrated System \leftrightarrow Test cases” bundled into the integration process.
 - Collaborative effort required for the definition of a common configuration & system management layer
 - Common Integration repository & bug tracking system
 - Efficient and fast submission / feedback process



Thank You



*Who hastens too much at the beginning,
comes to the end too late*

Plato

