

Project Status Report

Andrew Herbert



Workpackages

- Mobile Object Workbench
- Information Space
- Autonomous Agents Framework
- User Access Component
- Pilot Applications: ETEL++ and Bavaria Online

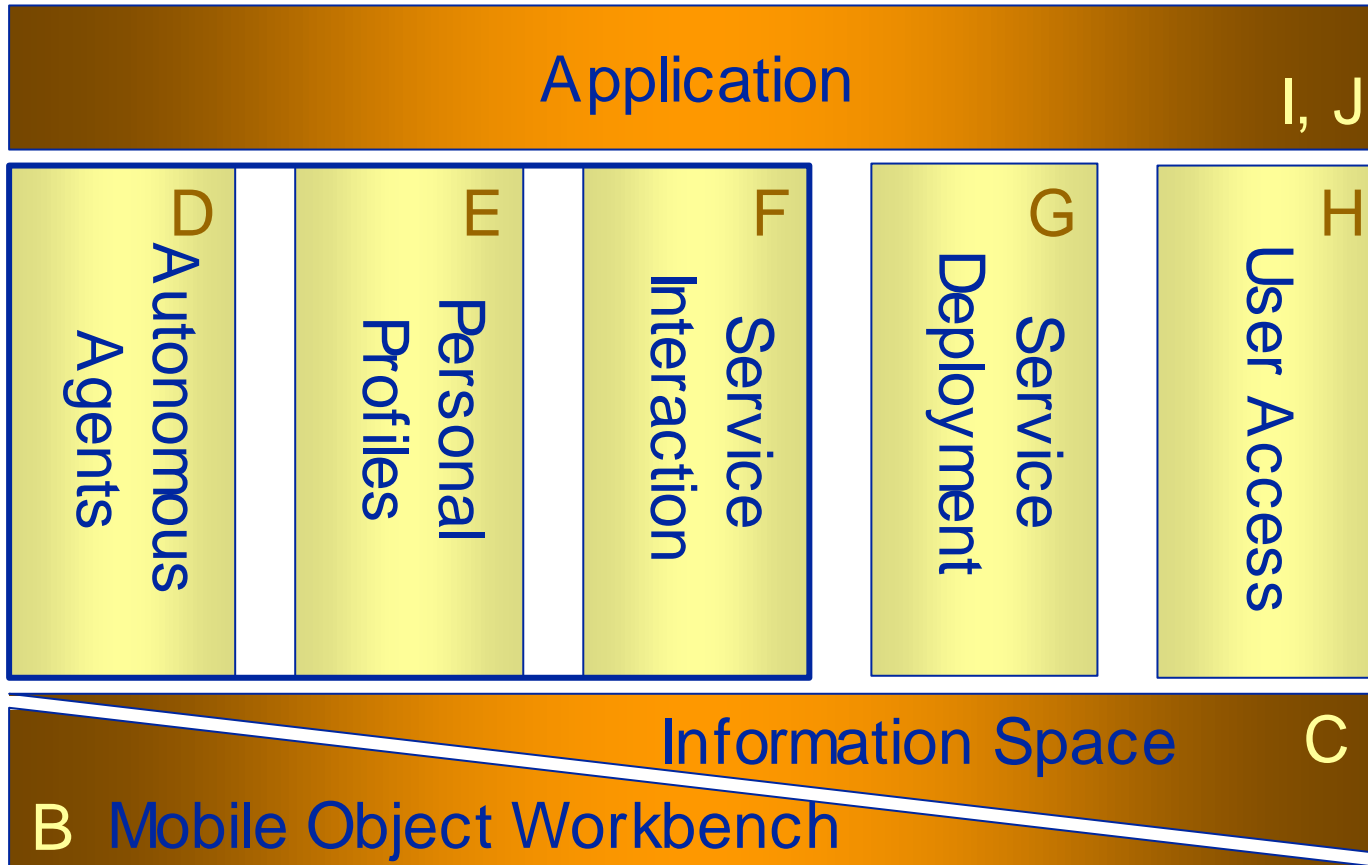


Project Results

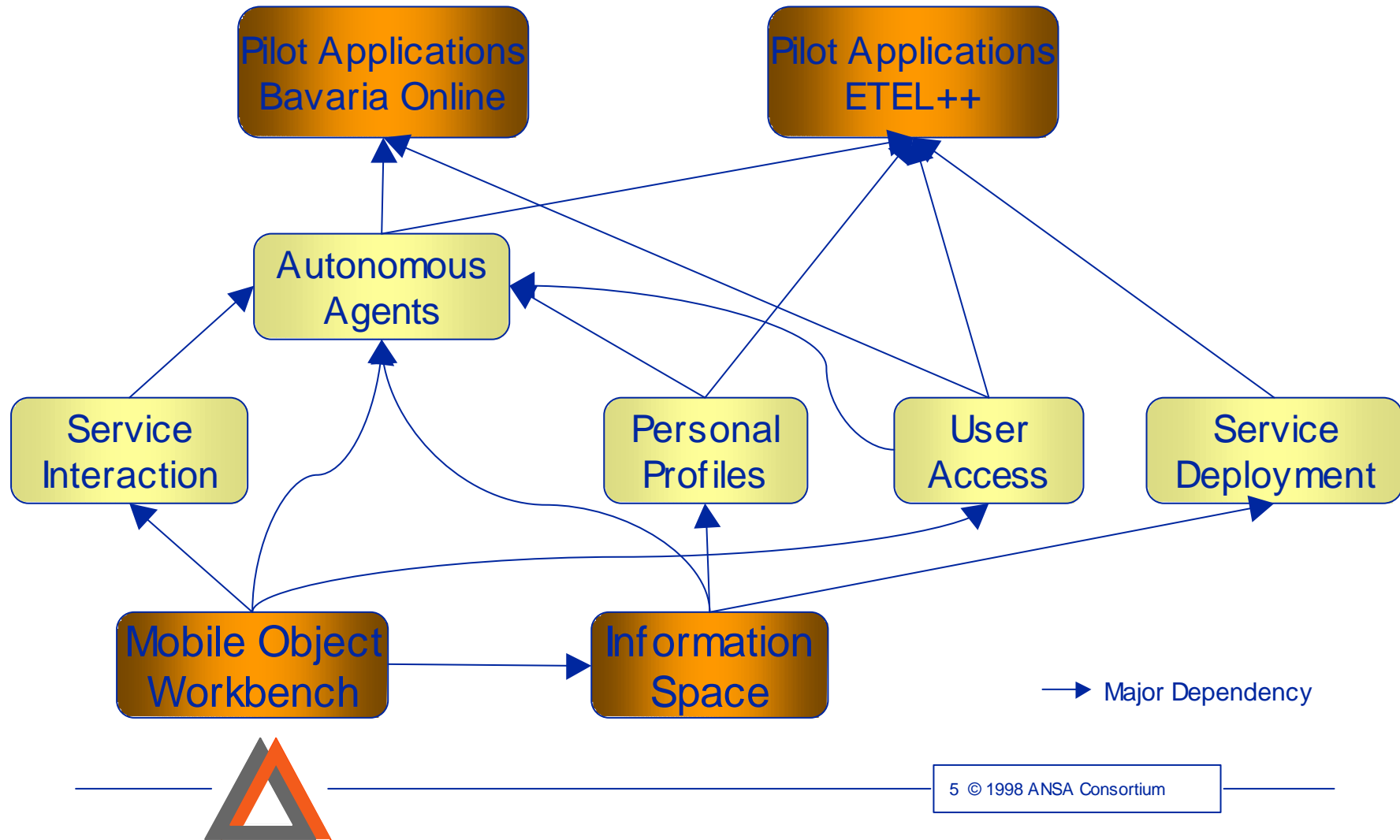
- Architecture for distributed mobile applications
- Infrastructure prototype
 - mobile object workbench / information space
 - agent framework
 - service deployment
 - user access
- Two pilot applications
 - FAST: Internet services through Bavaria Online
 - INRIA/TCM: Personalised newspaper
- Public report on architecture, user needs, implementation and pilots



WP A: Architecture



Component Dependencies



WP A: Architecture

- Co-ordinator: APM
- Major Results
 - Architecture DA1.1
 - repository for all work packages, focal point for inter project discussion and communication
 - Architecture DA1.2
 - component architecture based on the design patterns of all work packages



WP B: Mobile Object Workbench

- provides facilities ...
 - for location transparent references
 - to move objects
 - security features
 - to access local information
- is scalable
- Implementation based on Java/Flexinet



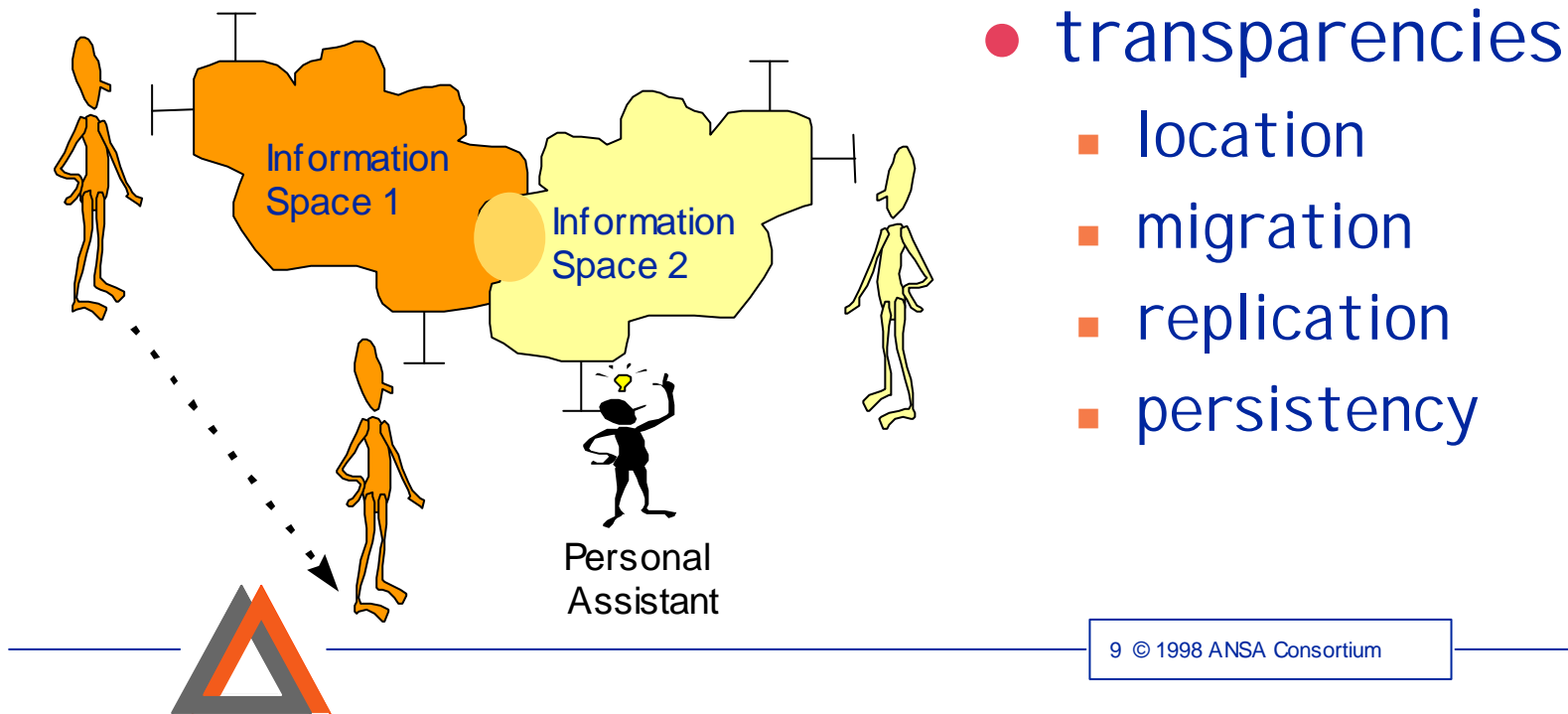
WP B: Mobile Object Workbench

- Co-ordinator: APM
- Major Results
 - Mobile Object Workbench V1.1 available (DB7.2, DB8.2)
 - extensions planned for security and object locator



WP C: Information Space

- Allows the user / mobile agent to maintain a consistent logical view to information



WP C: Information Space

- Co-ordinator: APM
- Major Results
 - Information Space Req., Design, Interfaces, Report (DC1, 2, 3, 6.1)
 - Implementation of Black-Box-Model available
 - Implementation of White-Box-Model to be done

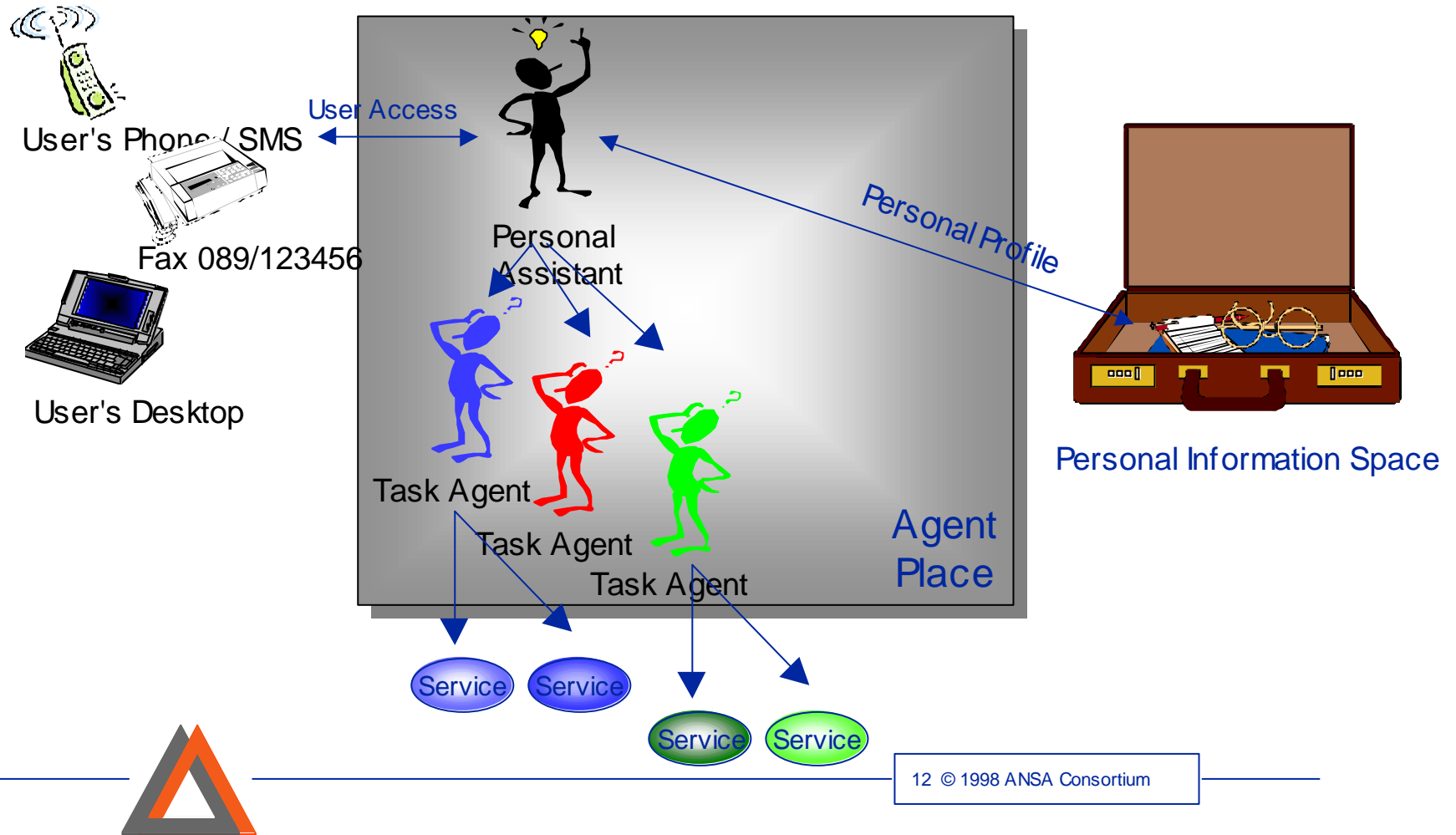


WP D, E, F: Agent Framework

- Components
 - Autonomous Agents
 - Personal Assistant
 - Task Agents
 - Personal Profile
 - Personal Data
 - Diary, Connectivity Information
 - Service Interaction
 - Flexible Interfaces to distributed services



Agent Framework



WP D: Autonomous Agents

- Co-ordinator: UWE
- Major Results
 - Design (DD 3)
 - Implementation started
 - Delay due to intensive interaction with Pilot Application
 - Corrective Action: Additional man power



WP E: Personal Profiles

- Co-ordinator: UWE
- Major Results
 - Design (DE 3)
 - Software (DE 5.1)



WP F: Service Interaction

- Co-ordinator: UWE
- Major Results
 - Design (DF 3)
 - Implementation started
 - Delay due to correlation with agent work package
 - Corrective Action: Additional man power



WP G: Service Deployment

- Monitoring of Performance
 - CPU, Disk, memory, network resources
- Data Mining Techniques
- Load Balancing Policies based on
 - user's mobility
 - supported output formats
 - user preferences
 - underlying architecture



WP G: Service Deployment

- Co-ordinator: Inria
- Major Results
 - Design and API (DG 3)
 - Implementation under way



User Access

- support for on-line and off-line access (e.g. fax, phone, SMS, www-browser, ...)
- Java-enabled and non-Java-enabled devices
- generic mark-up language and layout
- interfaces to adapt the quality of service to system parameter (network load, cpu, ...)



WP H: User Access

- Co-ordinator: FAST
- Major Results
 - Design & API (DH 3/4)
 - Implementation available
 - Device Gateways for Fax, http, Email available
 - Extensions planned for SMS



Pilot Application in Bavaria-Online

- Goal:
 - Demonstrate Use of FollowMe for Service Providers, Content Providers and Users
 - provide a prototype for a long-term service
- Two Pilot Services:
 - Portfolio Management System
 - Regional Event Notification System
- in co-operation with the Bavarian Citizens Networks



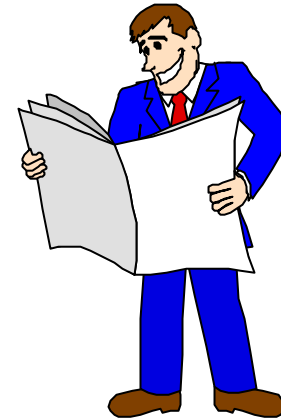
WP I : Pilot Application Bav. Online

- Co-ordinator: FAST
- Major Results
 - Requirements (DI 2)
 - Architectural Design available



Pilot Application: ETEL++

- ETEL++: A personalised newspaper
 - presented at a defined quality of service level
- Goal: demonstrate the need
 - for mobile code and data
 - for multiple terminal support



WP J: Pilot Application ETEL++

- Co-ordinator: Inria
- Major Results
 - Requirements (DI 2)



Summary

- The Project had a quick start-up and is now under full operation
- The Architecture builds a backbone of the project and provides the glue between components
- Individual SW components are available as planned
- Time slips in the implementation of Agents and Service Interaction are recognised and under control



Architecture (DA1)

Will Harwood



Why Architecture?

- “If builders build buildings they programmers write programs then civilisation would be destroyed by the first woodpecker.”
 - Gerald M. Wienberg
- “The purpose of architectural plan is to create an elegant building not an elegant piece of paper”



Objectives

- To produce a coherent articulation of the FollowMe architecture
 - to promote the use of common solutions to problems within the project
 - to capture our understanding and experience so that it doesn't die with the project



The Task

- Evolving series of architecture reports
 - (month 2) block diagram architecture
 - (month 6) proposed patterns architecture
 - (month 12) final architecture
- Steer design and implementation tasks
 - componentisation
 - stable interfaces
 - minimum duplication
- Use APM ISO Reference Model for Open Distributed Processing as a starting point

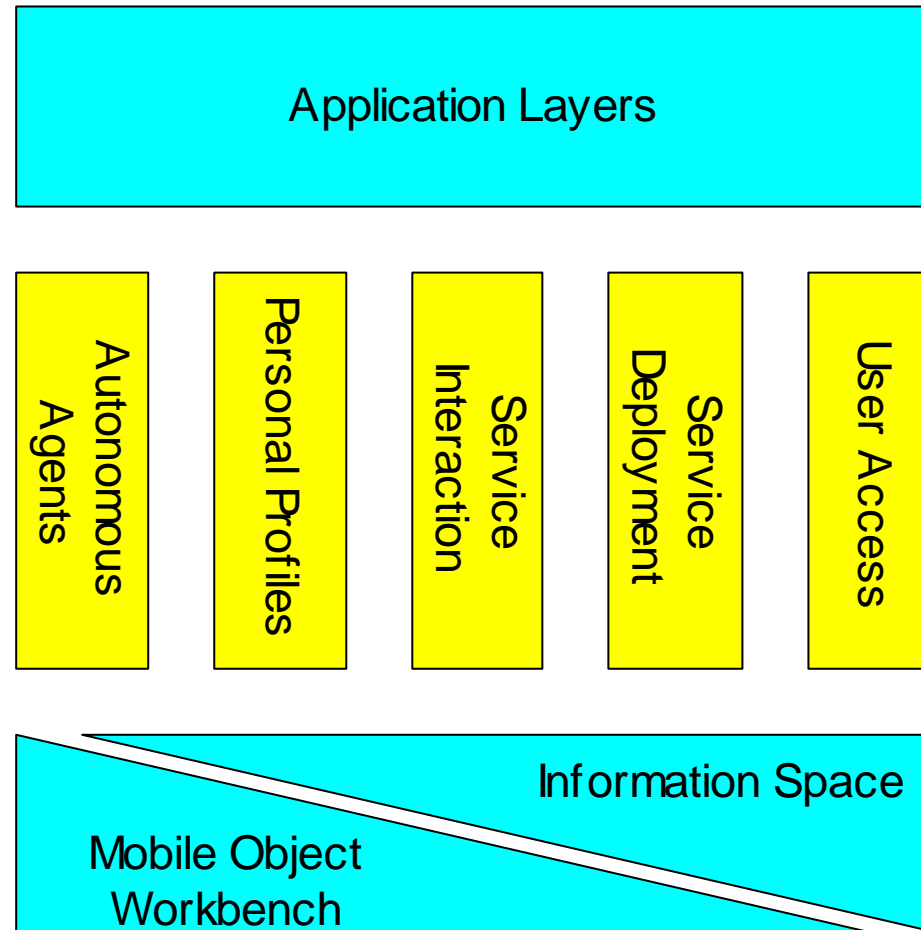


DA 1.2

- The general FollowMe framework
- The basic technologies
- The design patterns



The FollowMe Framework



Foundations

- Basic infrastructure for “Internet Objects”
 - portable
 - run on any client or server platform in the Internet
 - mobile
 - carry intelligence with data
 - provide a visitors room rather than having to export internal functions for ease of security and management
 - protocols - e.g. payment
 - persistent
 - can leave information in the system
 - mask storage replication and storage access methods
 - dynamically distributed
 - allow dynamic determination of client/server split



Reusable Components

- User Access
 - talk to the user in a device independent way
- Service Deployment
 - monitoring infrastructure for dynamic service configuration
- Personal Profiles
 - ubiquitous model for information about user
- Service Interaction
 - templates for using services to enable scripting
- Autonomous Agents
 - take on tasks on behalf the user



Technology Base

- ISO Reference Model for Open Distributed Processing
 - object-oriented modelling framework
 - separation of concerns
- CORBA
 - global consensus that WWW is evolving to Distributed Object Computing
 - OMG CORBA dominate D.O.C. standards
- Java
 - widely adopted language for D.O.C.
 - bytecodes model enables mobility
 - fits ODP conceptual framework better than C++
 - absorbing CORBA protocols
 - reflection and introspection give cleaner APIs



Patterns

- “Each pattern is a three-part rule, which expresses a relation between a certain context, a problem, and a solution” [*Christopher Alexander*]. Moreover each pattern:-
- “...is the **abstraction** from a concrete form which keeps recurring in specific non-arbitrary contexts” [*Dirk Riehle and Heinz Zullighoven*],
- Patterns are a means of piecemeal growth



Properties of Patterns

- expresses an **encapsulation** of a separable aspect of an overall design task,
- expresses a balance or **equilibrium** between opposing requirements,
- is **open** to extension and modification,
- is **composable** with other design patterns,



Describing a Pattern

- **Name**
- **Context** A description of context for the use of the pattern.
- **Problem:** A description of the conflicting forces to be resolved and constraints placed upon the resolution, and how these interact with one another.
- **Solution**



Two Pattern Hierarchies

- Patterns may **refine** other patterns
 - Patterns may multiply inherit
- Patterns may **use** other patterns

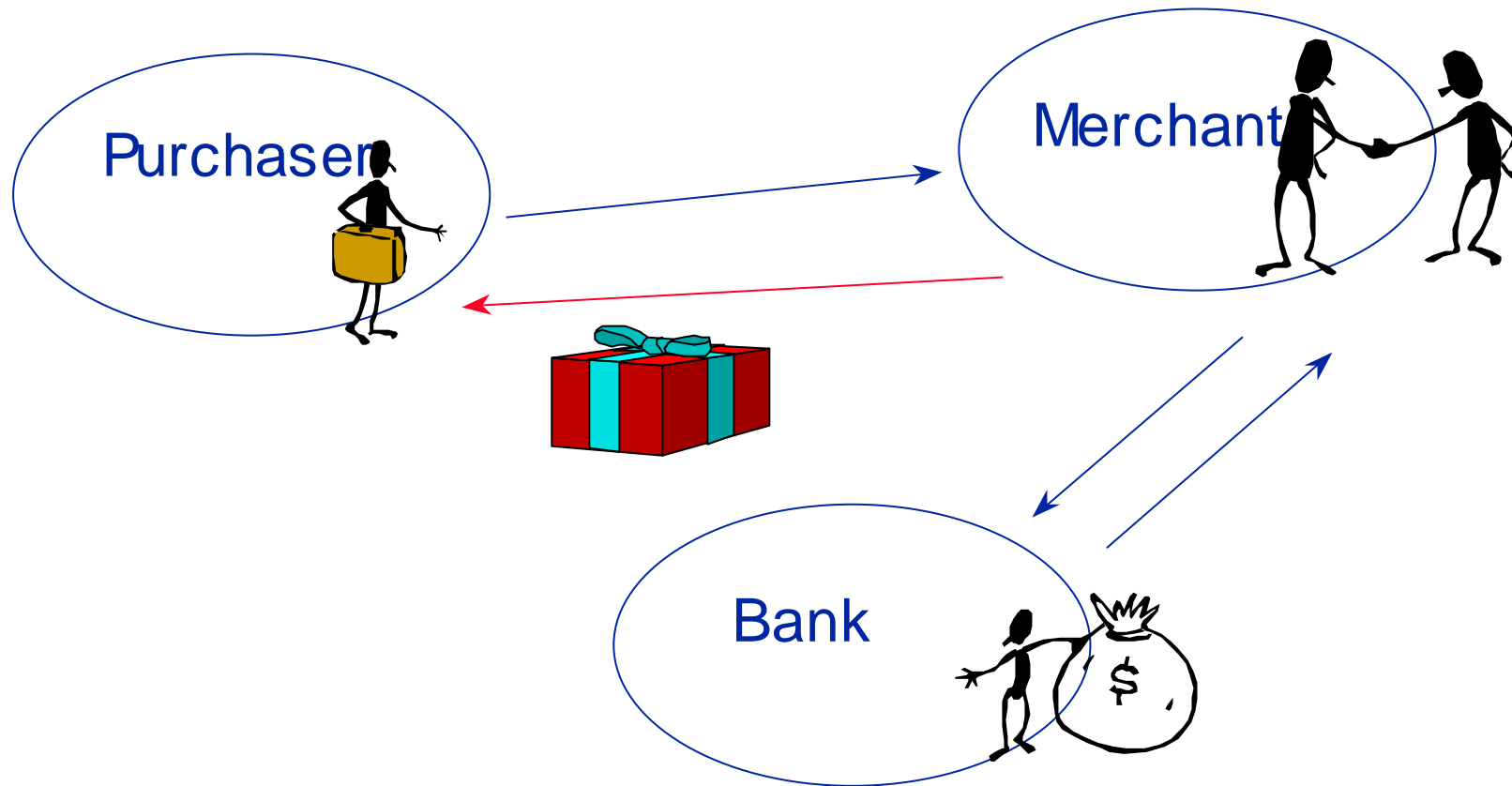


Basic Patterns

- Objects
- Collective objects
- Object clustering
- Meta objects
- Proxying
- Flavouring
- Reactive object
- Active object
- Autonomous object
- Location transparent reference
- Dynamic binding
- Server
- Factory
- Introspection
- Self-Interpreting data
- Execution vehicle
- Monitored object
- Deployable
- Protocol object
- Service Map
- Service Directory
- Agent



Example Pattern Protocol Objects



What Next?

- Dissemination
 - SIGOPS 98, Middleware 98, Mobile Agents 98
- Add patterns as work develops
 - look for generics
- Monitor integration of components
 - ensure interfaces are architectural
 - ensure foundations are adequate
- Monitor industry developments for insight, alignment and opportunities
 - OMG MASIF, Aglets, Voyager, others

