

FlexiNet

Overview & Plan for 1997

Richard Hayton



Current Situation

- Increased use of Distributed Object Technology
 - CORBA, RMI, web, agents ...
- Increased diversity of applications
 - new & different requirements
 - network aware applications
- Rapidly changing environment
 - application code in client, server and switches
 - applications require more from network
- Rate of change is INCREASING



The Problem

- Current implementations are slow to react
 - panacea approach cannot evolve
 - standards bodies take time
 - no evolution between versions
- RISK
 - platforms will fail to evolve
 - need to regularly change to new platform
 - platforms evolve beyond design assumptions
 - become inefficient and hard to use



Solution

- Build to evolve
 - assume design assumptions will change
 - design for multiple instances/versions
- FlexiNet
 - soft, dynamic, modular ODP platform
- Application driven abstractions
 - abstractions are tools that are downloaded into hosts and switches as required
 - no central coordination of tool design
- Blur distinction between application and platform



Benefits

- Unified application and network management
 - simplifies design and management
 - addresses end-to-end QoS
- Handles legacy and evolution
 - existing applications continue to work
- Optimize for today, evolve tomorrow
 - don't need to second guess future bottlenecks
- Quicker to market, less expensive to maintain
 - easy to extend to meet new technologies



Focus

- Three areas identified by FlexiNet workshop
 - FlexiNet scenario
 - demonstrate feasibility
 - example of look & feel
 - dependability
 - generic support & specification
 - multimedia
 - explicit binding
 - resource awareness



Project Goals

- Identify & specify key infrastructure
 - determine framework for modularity
- FlexiNet framework
 - core meta-abstractions to allow evolution
 - example: support for multiple binding models
- Investigate current ODP Issues
 - support for MultiMedia
 - explicit binding, reservation/adaptation
 - user defined signalling
 - network awareness



FlexiNet Architecture

- Nucleus
 - “micro-kernel” approach
 - component model
 - common abstractions
- Modules
 - unit of software management
 - dynamically loaded
 - configurable and reflexive
 - evolvable



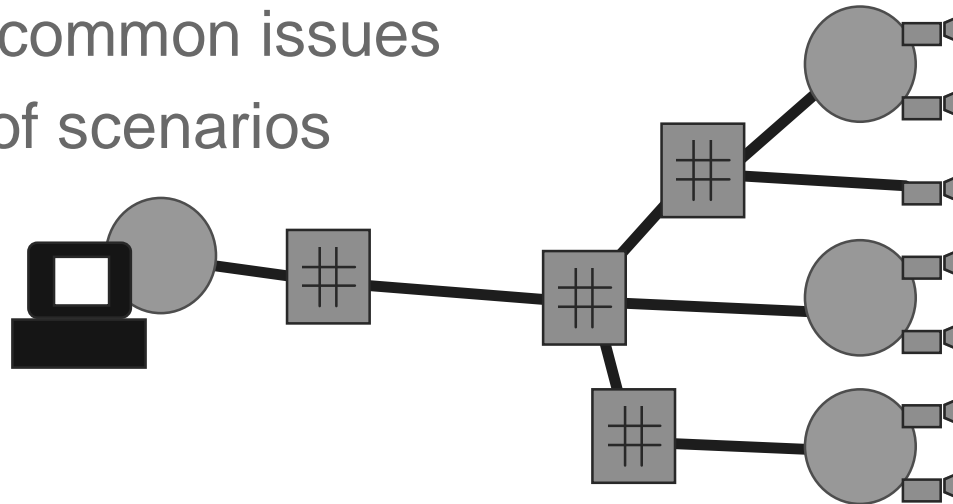
Nucleus

- Binding model
 - multiple ‘competing’ binders
 - recursive binder specification
 - implicit and explicit binding
- Framework for QoS/Dependability specification
 - manage choice of binder
 - manage choice of QoS
- Nucleus gives us the connectors to plug and play with modules



Work Package 1: FlexiNet Scenario

- Based on surveillance camera example
 - illustrates many common issues
 - in a wide range of scenarios



- Describe the components and interfaces
- Develop a prototype nucleus
- Implement / simulate the modules required

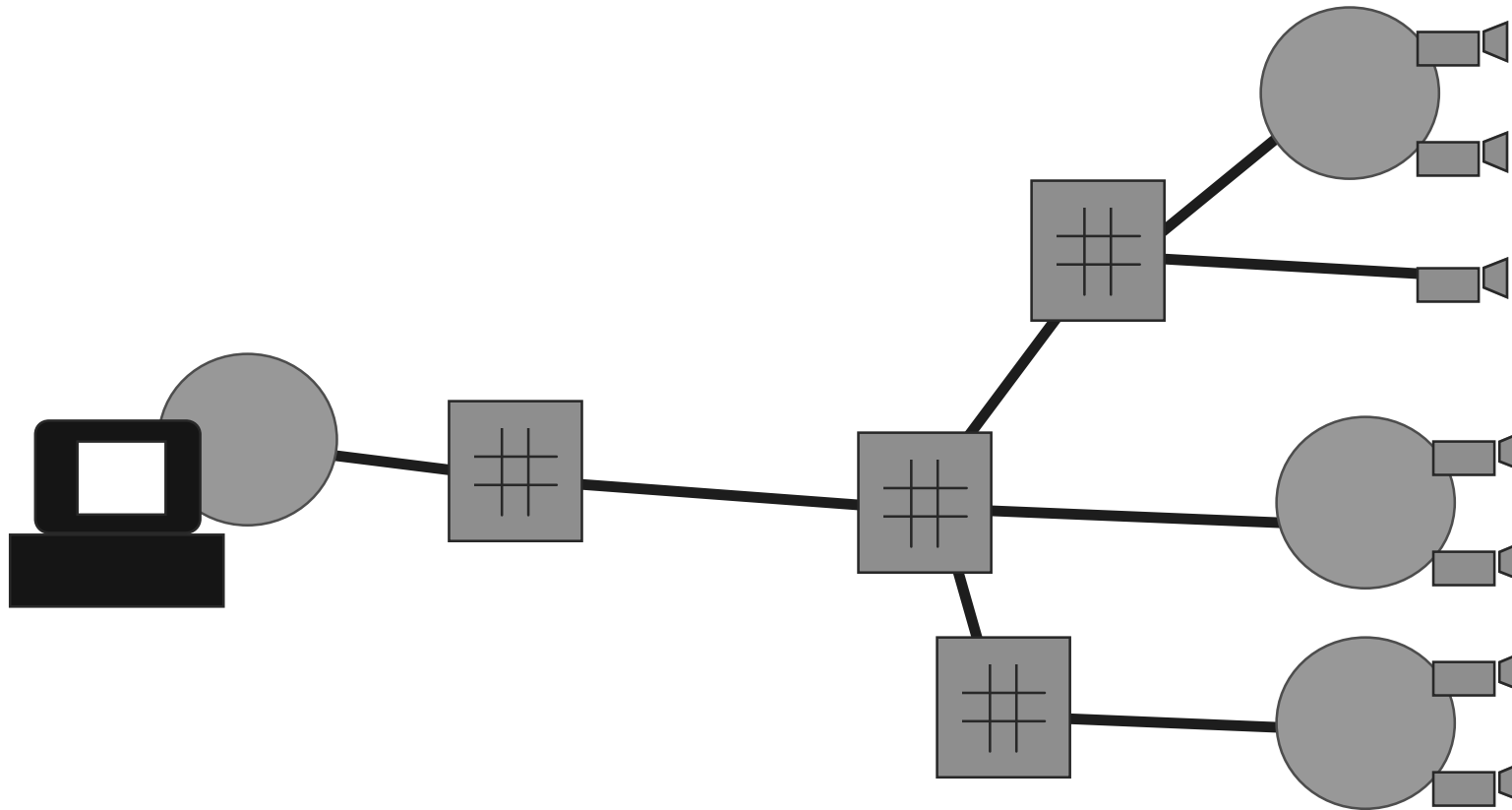


User Defined Signalling

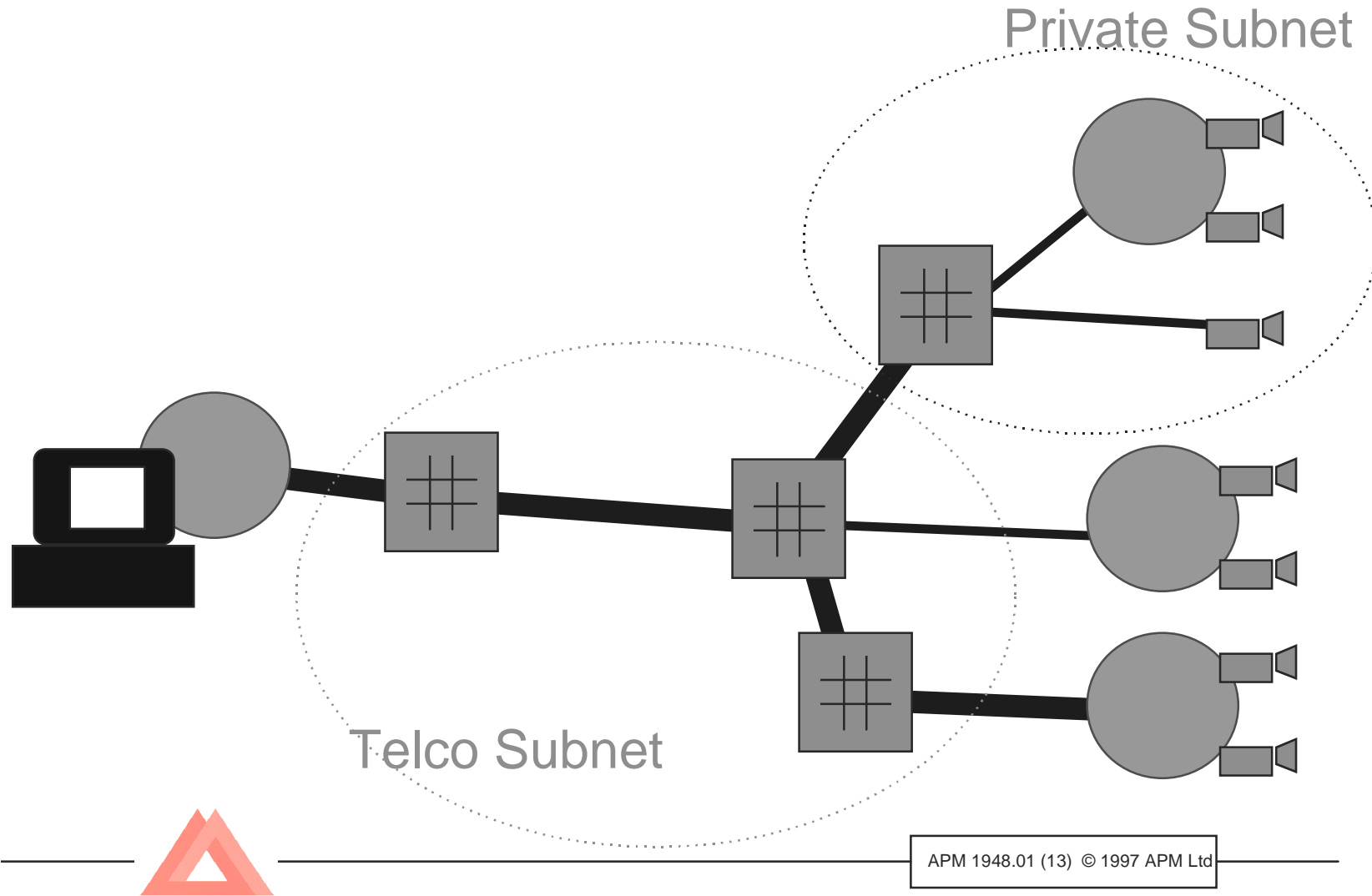
- Complex signalling requirements
 - application specific
- Applications must be “Network Aware”
 - make efficient use of network
 - limit unnecessary signalling
 - processing within the network



User Defined Signalling Example



Physical Interconnect

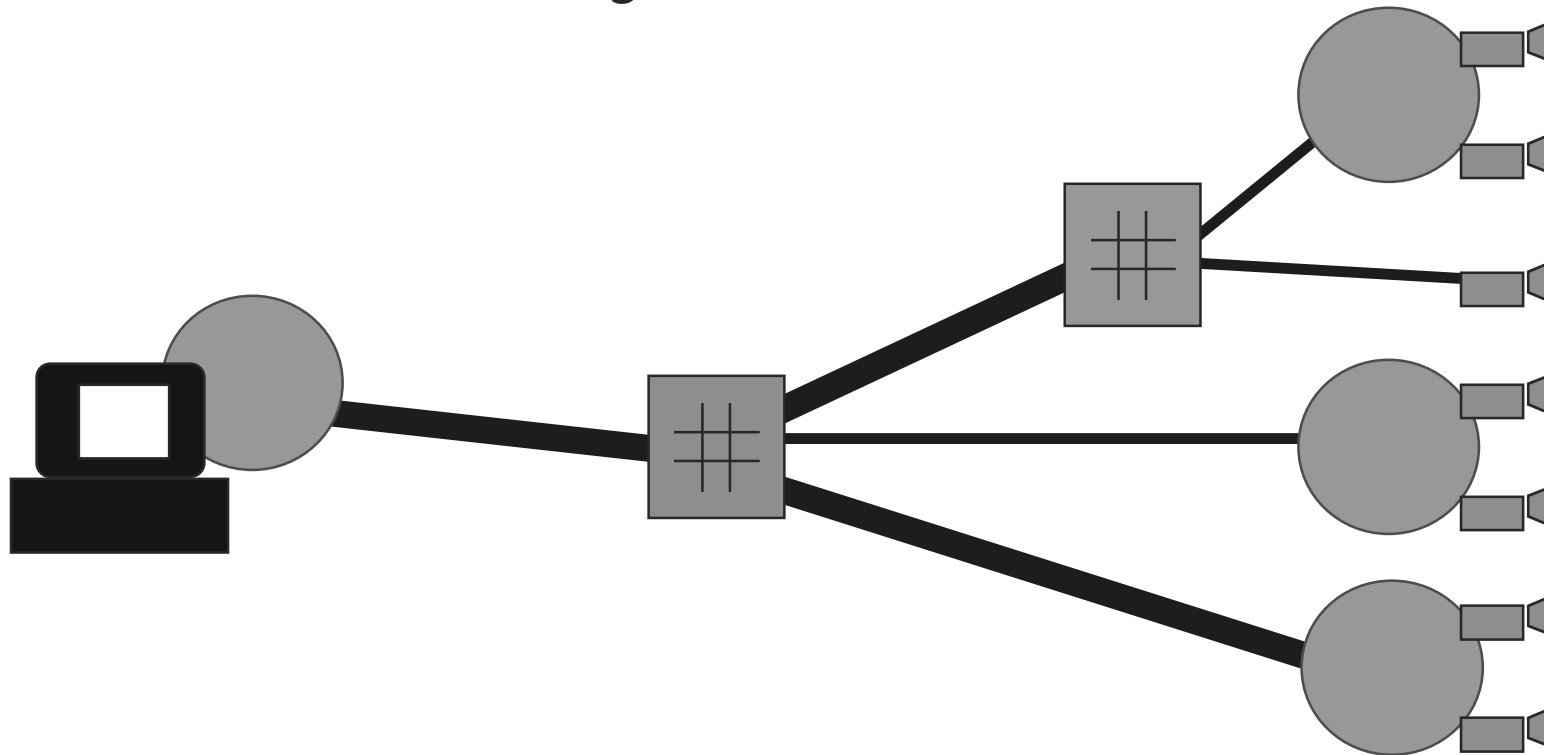


Private Subnet

Telco Subnet

Virtual Networks

- Abstraction to support used defined signalling
- Model *interesting* features of network



WP2: Modular Dependability

- Specification and control of nonfunctional aspects
 - concurrency control, atomicity, replication, check pointing, persistence, security
- Investigations:
 - how practical is “plug & play” dependability
 - how best to present reflexive interfaces
- Detailed design & implementation of nucleus



WP 3 - Interactive Multimedia

- Build on experience from DIMMA, DCAN, ReTina
- Continue and extend work on explicit binding
 - exploit new models
 - group communication
- Investigate new technologies
 - virtual networks
 - resource aware kernels
 - reactive languages



Summary

- The world is, and always will be changing
 - First to market wins
- Applications demand more
 - tighter controls on resources
 - large “toolbox” of abstractions
 - greater awareness of environment
- FlexiNet gets you there.



Deliverables

Deliverable	Due End
Report on scenarios and the benefits and issues they raise	March 97
Proof of concept demonstrator	May 97
Architecture for modular dependability report	May 97
QoS investigation report	May 97
Implementation plan for modular dependability	June 97
Micro kernel investigation report	July 97
QoS management Abstractions	July 97
Definition of binding language and framework	August 97
Binding prototypes	September 97
QoS integration of bindings and resources prototype	October 97
Representative set of key meta classes prototypes	November 97
Predictable modules prototypes	December 97
Demonstration application	February 97
Evaluation report	March 1998

