



---

**Poseidon House  
Castle Park  
Cambridge CB3 0RD  
United Kingdom**

TELEPHONE:  
INTERNATIONAL:  
FAX:  
E-MAIL:

**Cambridge (0223) 323010  
+44 223 323010  
+44 223 359779  
apm@ansa.co.uk**

---

## **ANSA Phase III**

# **Managing Dependability in the Electronic Market Place - Slides**

**Nigel Edwards**

### **Abstract**

This is a presentation of the paper APM.1203

This presentation shows how to integrate the management of a service's dependability with other aspects of service management including naming and trading. The intention is that service management can be specialized by building further services on top of the basic services described.

The underlying paradigm presented to the programmer is that each service has associated with it a management service. Clients use the management service during trading to find the right service instance and can appeal to the management should a service appear to be faulty.

The the motivation for this work is the need to make services dependable in the electronic marketplace. The presentation includes a case study in which the management services are used to manage a Distributed Information Publishing System (DIPS).

---

APM.1246.00.05

**Draft**

25 April 1995

Request for Comments (confidential to ANSA consortium for 2 years)

---

**Distribution:**

**Supersedes:**

**Superseded by:**

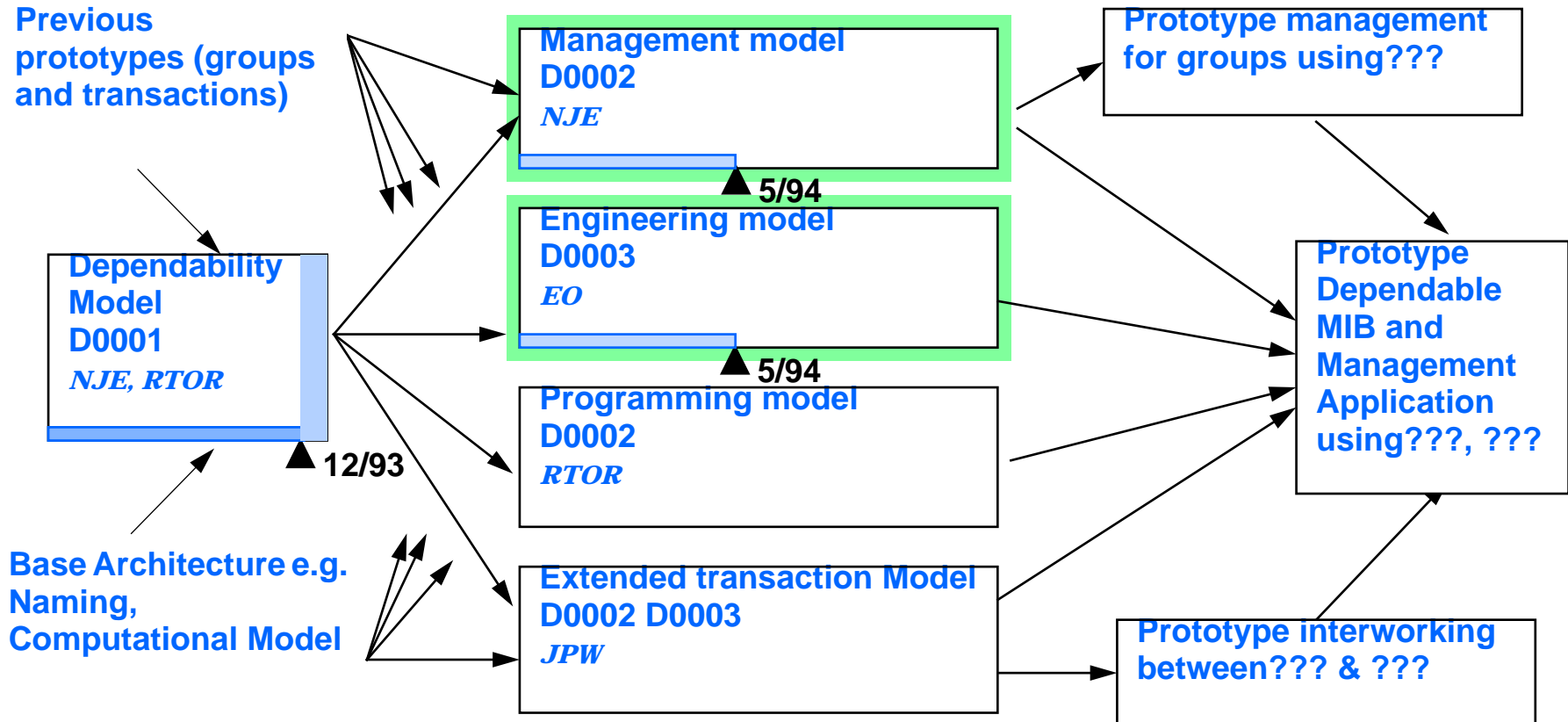




## Managing Dependability in the Electronic Market Place

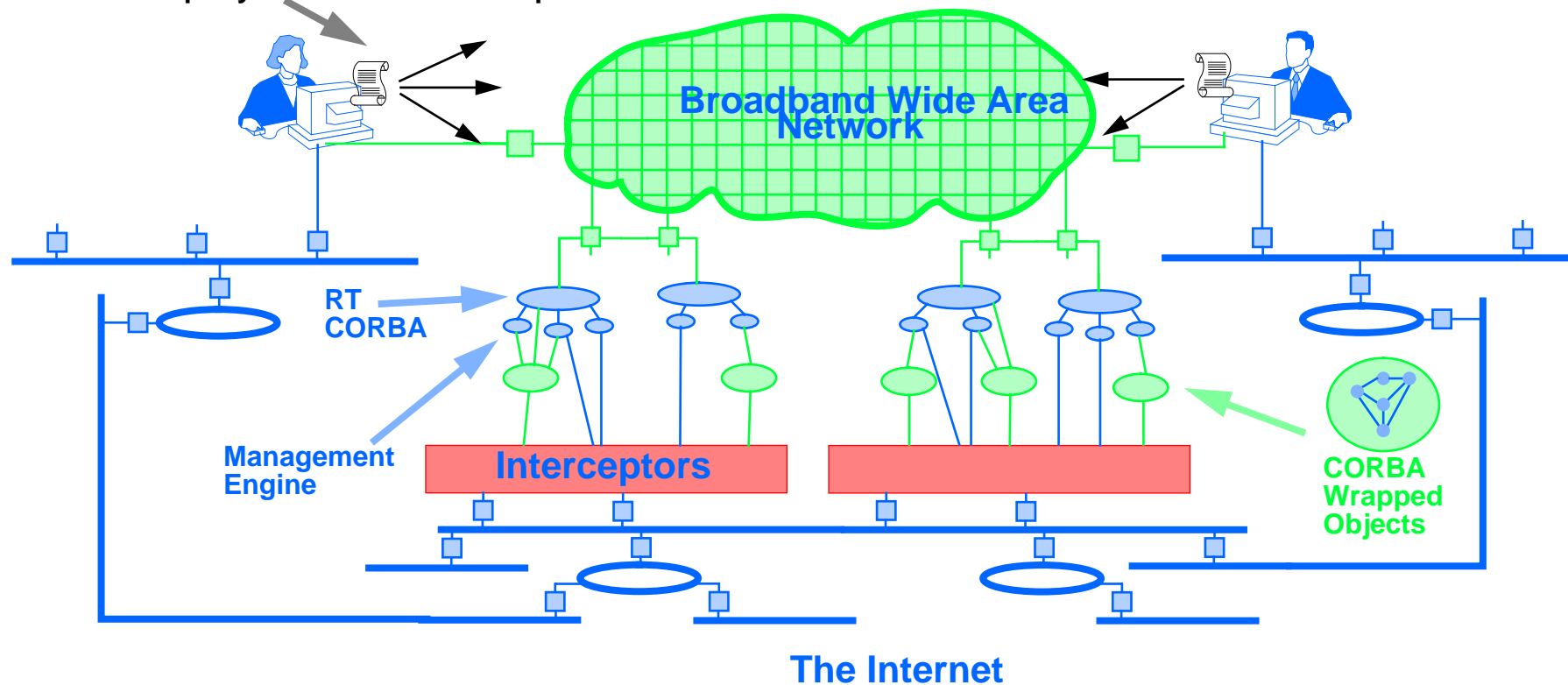
- **Objectives of this talk**
  - To explain the main concepts of the Management Engine for Dependability (MED)
  - To show how it can be applied to the electronic market place
  - To get your feedback on it
- **Caveat Emptor — these are very much initial, untested thoughts**
- **Summary**
  - Where it fits into the old workplan and the proposed work
  - Motivation — DIPS
  - Main components of MED
  - How it could be used in DIPS
  - Related work, conclusions & future work
  - More details in APM.1203

# Dependability plan



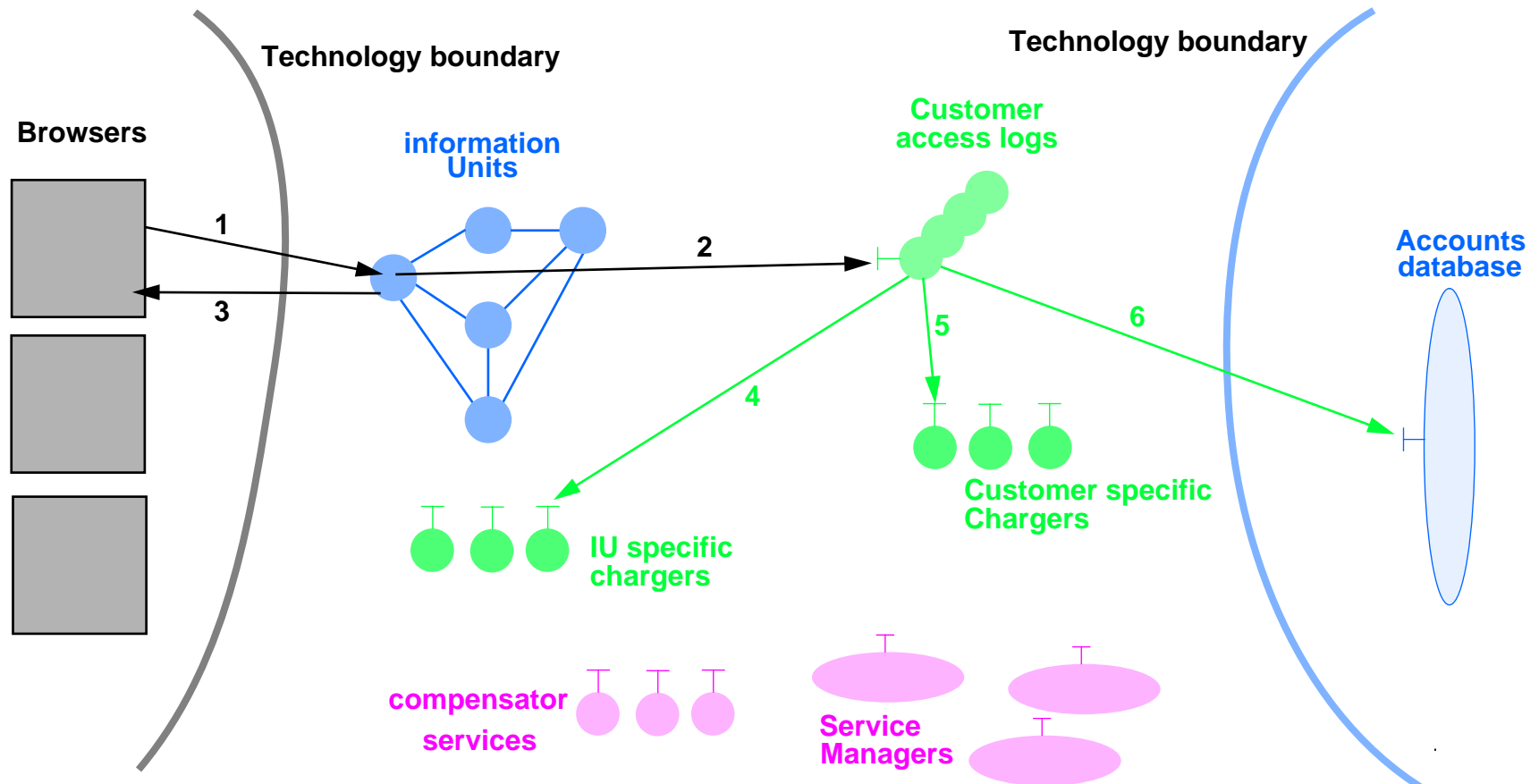
# The Dependable Electronic Market Place

Multi-party coordination via scripts

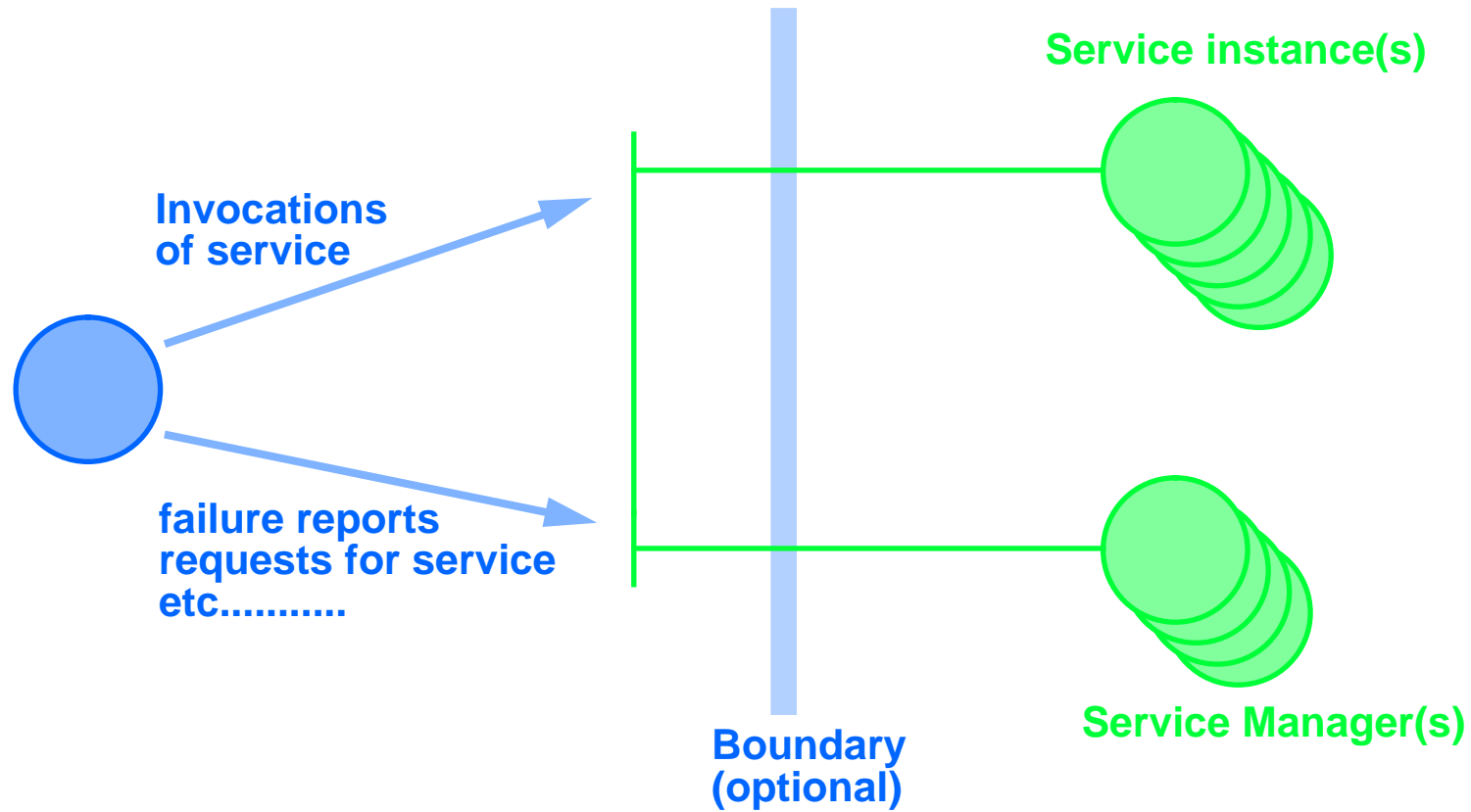




# DIPS: An Information Services Scenario (APM.1171)



## The management paradigm



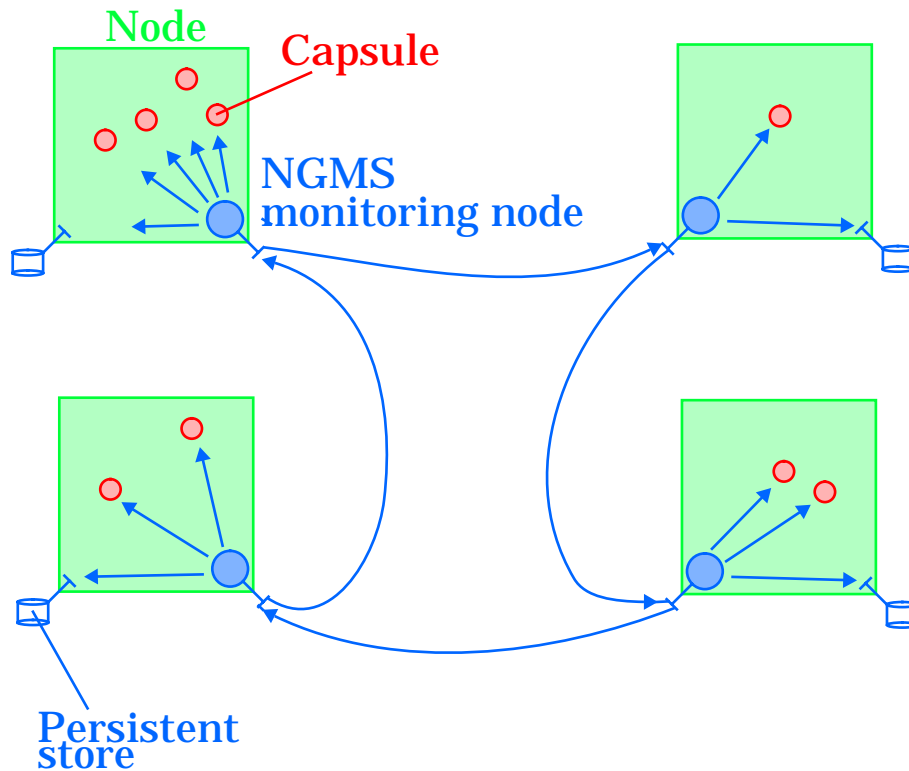


## Technical objectives for MED

- Failure detection without high overhead (minimal use of heartbeats)
- Failure arbitration
- Management of available instances
- Service instantiation on the most appropriate node
- Integration & support for the service lifecycle
  - activation/passivation
  - trading
  - naming
  - in service upgrade
- Failure assumptions:
  - MED Components: fail silent
  - Managed services: omission, timing, value (+ some types of consistency failure)

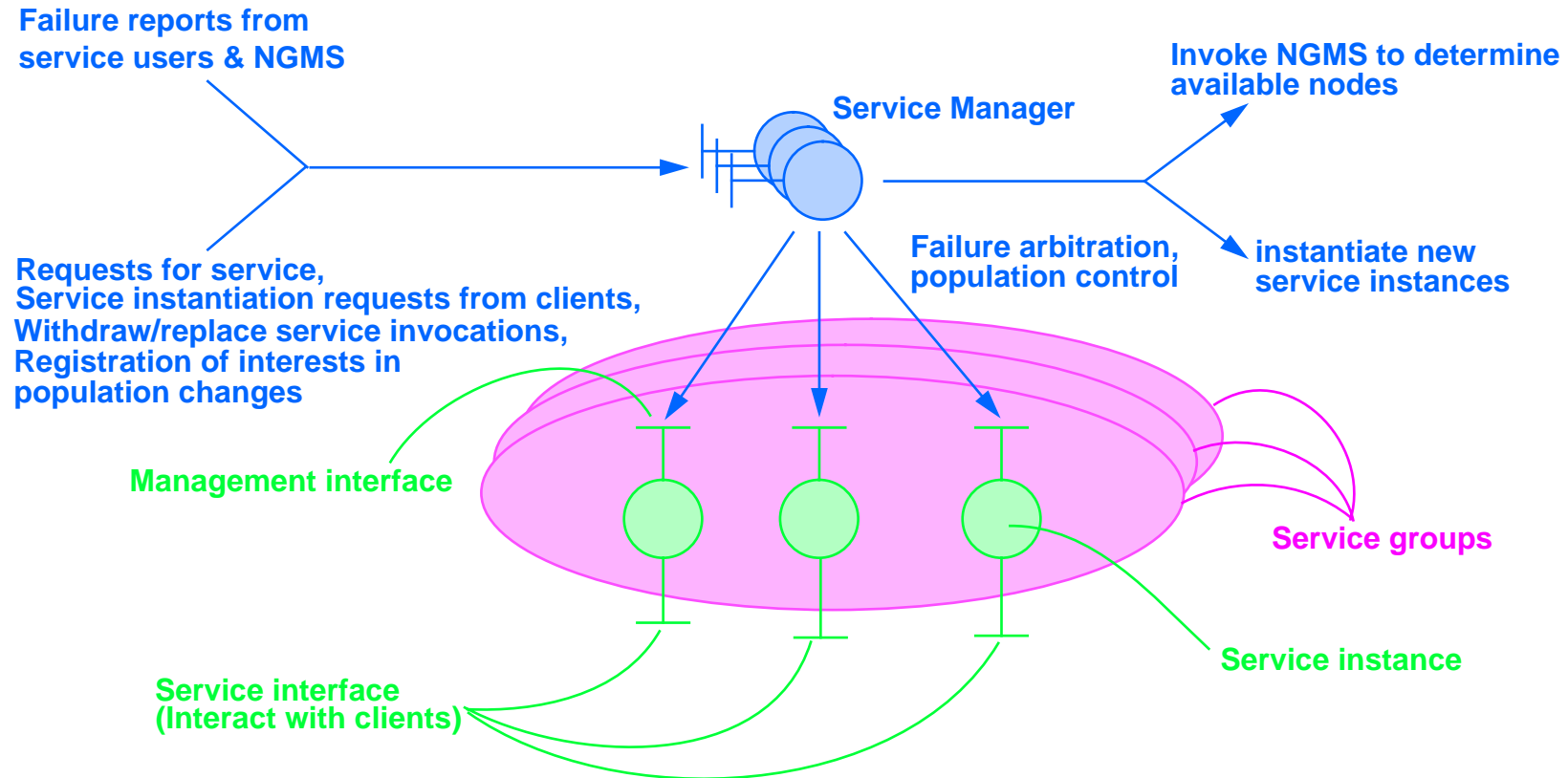


## Node Group Membership Service

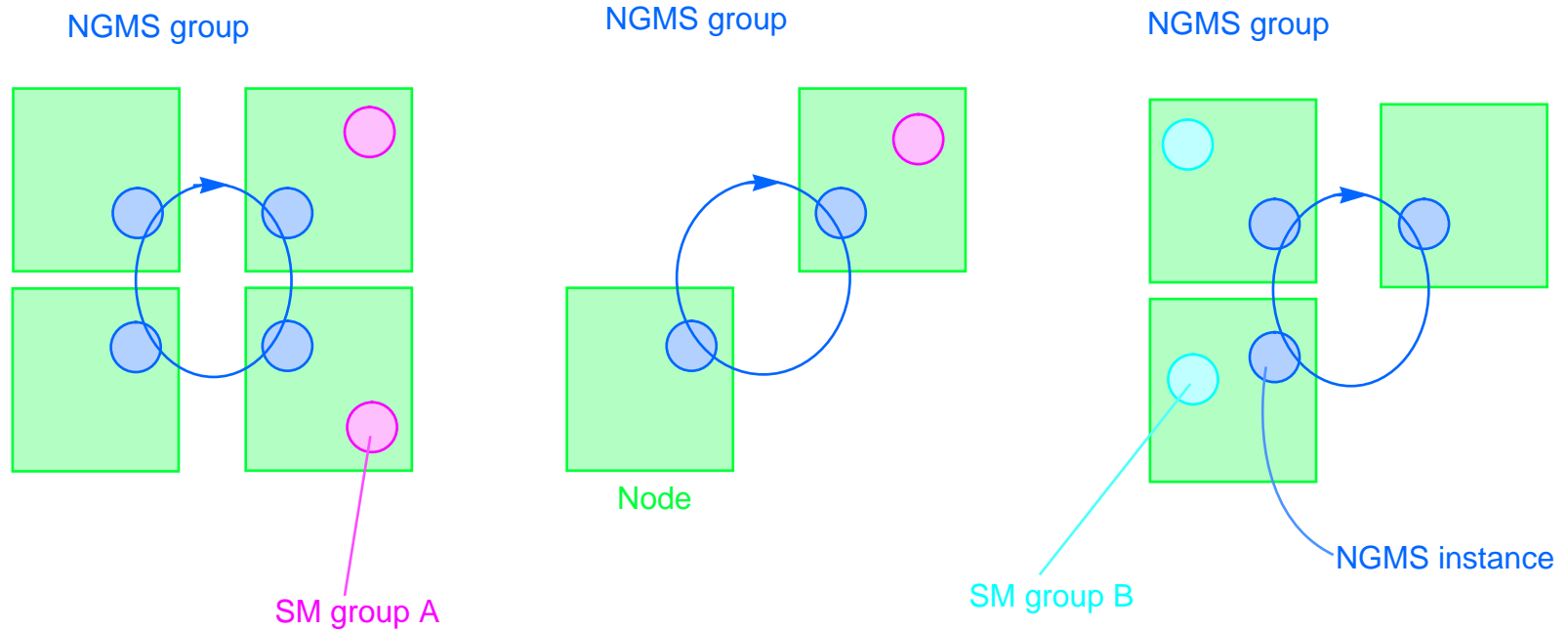


- Registration for death notifications
  - c.f. notification service
- Local detection of:
  - capsule death
  - store death
- Registration information is distributed
- Death of node — membership protocol
  - synchronous v. asynchronous

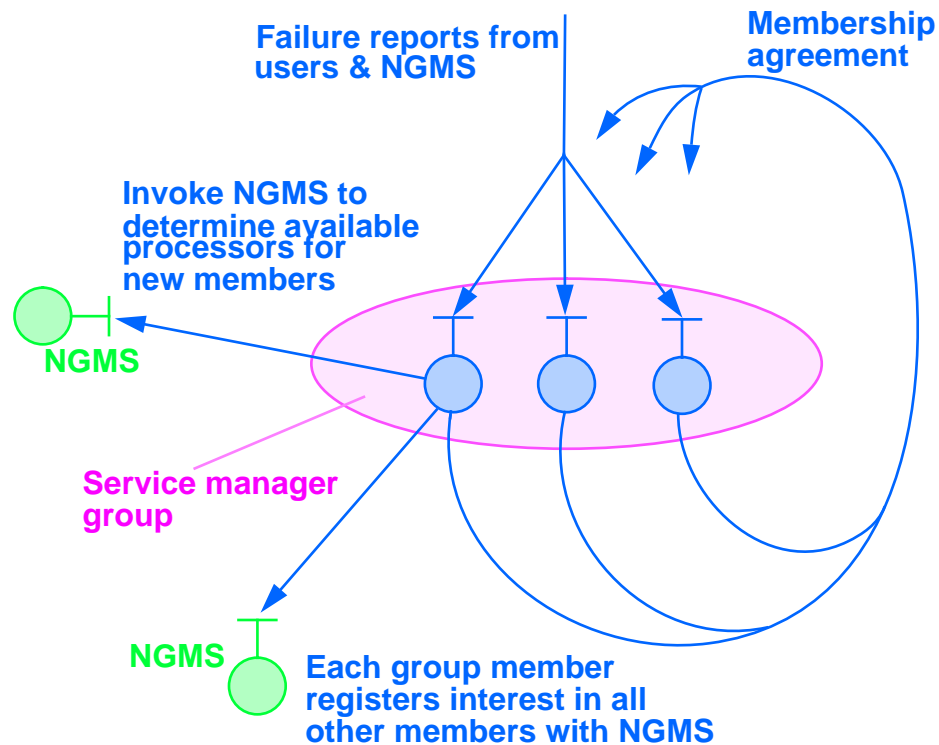
# The service manager



# Node Group Membership and Service Management Services

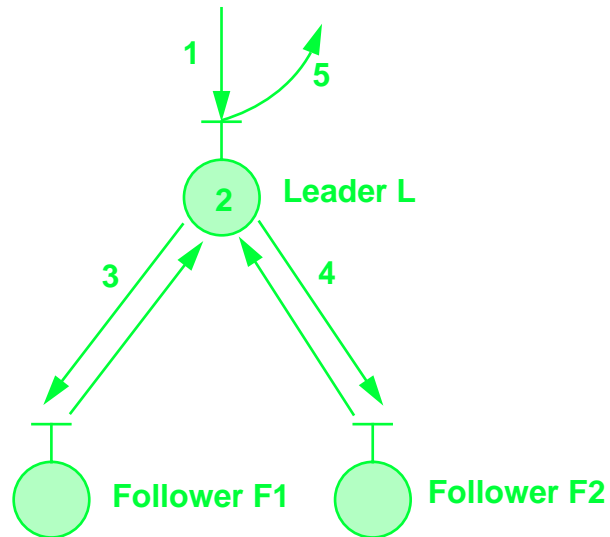


## Managing the Management Engine



- Service managers only communicate for failure reports (run a membership protocol) and updates
- Text based tool or (e.g.) OpenView to bootstrap NGMS & service managers
- Needs to be dependable => replicated files

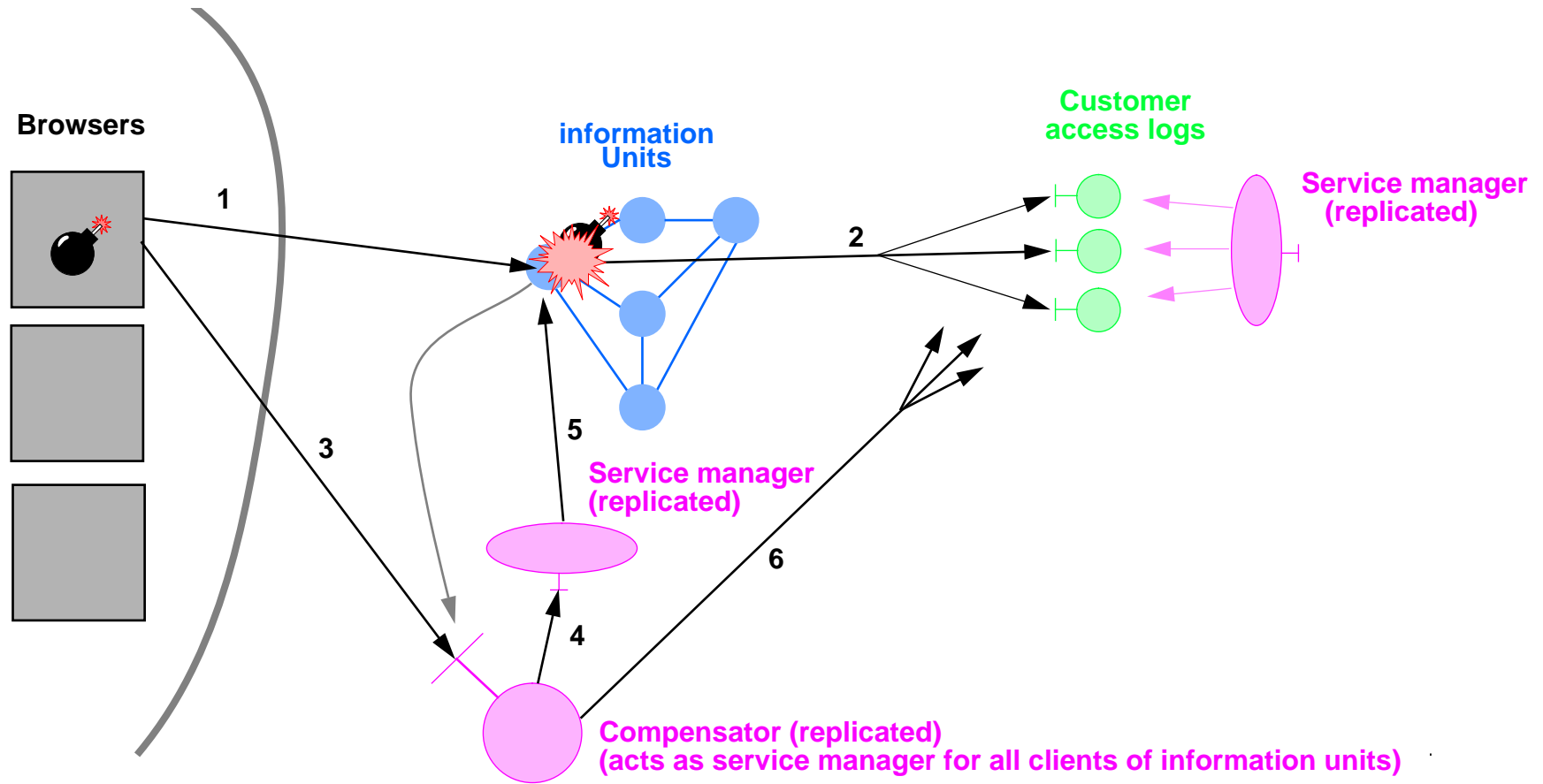
## Service Manager Group Update



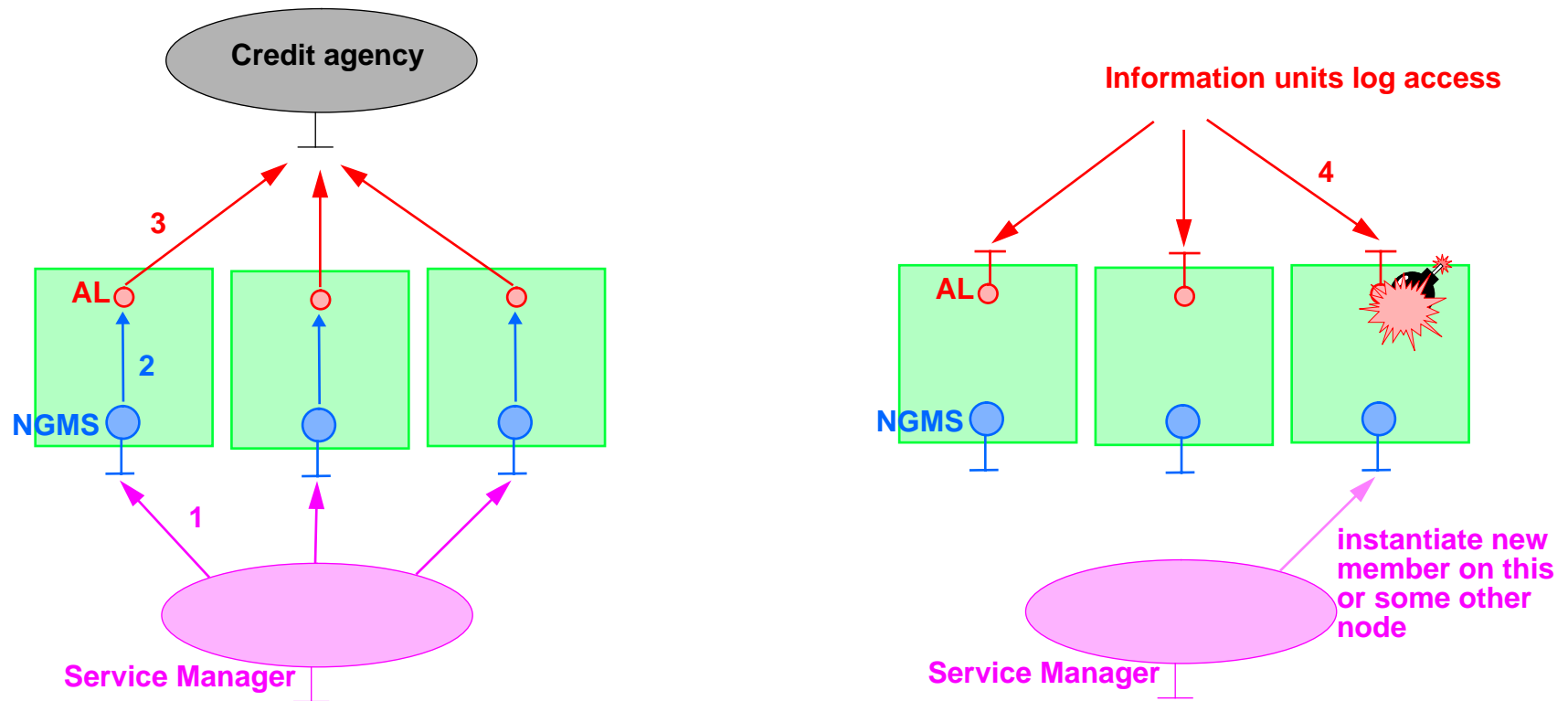
1. Client makes invocation
2. L executes operation (may invoke third parties to instantiate new services)
3. L sends update to F1
4. L sends update to F2
5. L sends reply to client

- **Only the leader invokes 3rd parties (NGMS)**
- **Failure recovery => query environment (NGMS)**
- **State updates idempotent — keeps recovery simple**
- **Membership protocol — the hard part!**

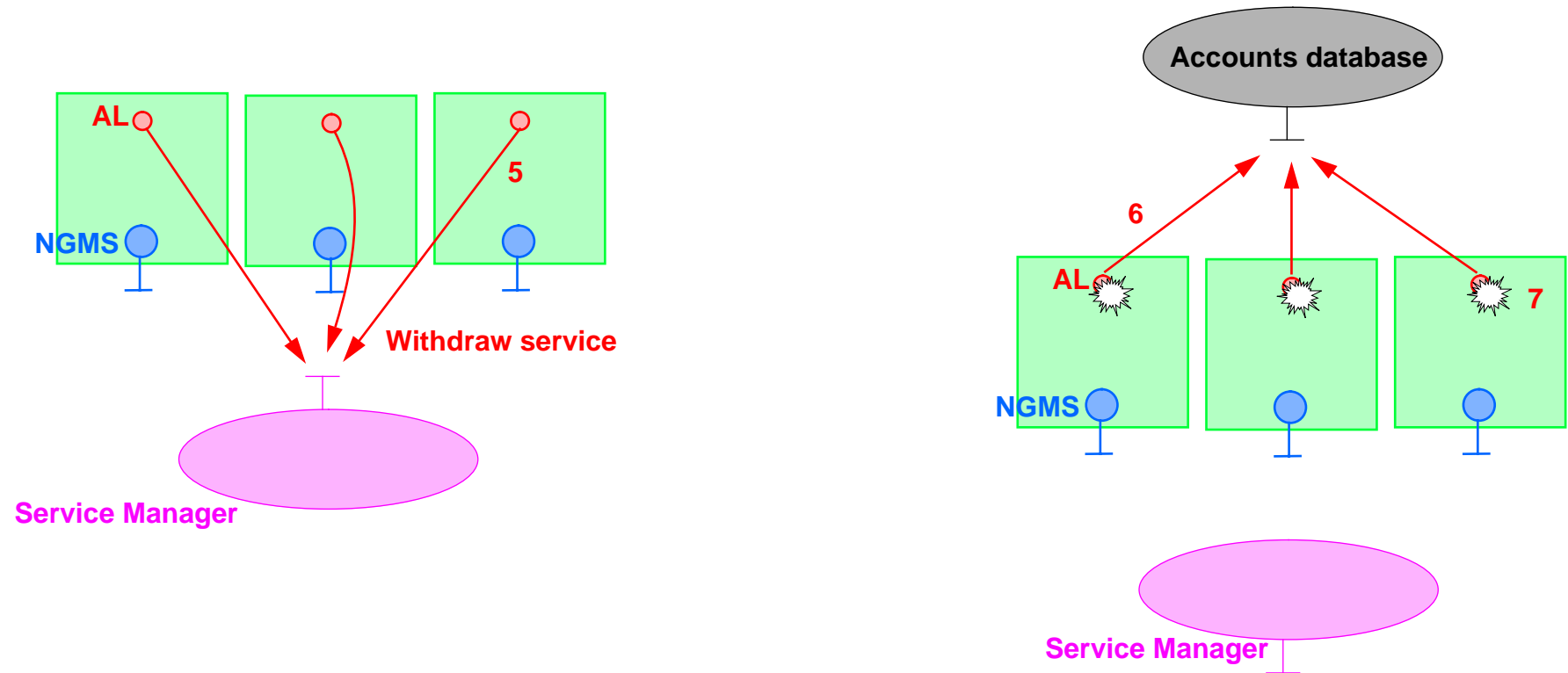
# Failures in DIPS



# The Access Log Lifecycle (1)



## The Access Log Lifecycle (2)







## And Finally.....

- **Influences: Isis, Cristian, Shrivastava, Delta-4 and previous ANSA work**
- **Contribution:**
  - Integration of fault management with lifecycle management
  - Scaling by splitting fault detection and management
- **Architectural consequences for trading, relocation**
- **As well as DIPS, MED could support: the Advanced Trader (APM.1177), Web Cache (APM.1204 & APM.1241)**
- **Extensions:**
  - Isolation and fault management of non-fail silent node
  - Naming and location of collections (see workshop report APM.1220)
  - Load balancing
- **But first.....  
We need to prototype and test the basic services.**