

Distributed Control Loop Patterns for Managing Distributed Applications

Ahmad Al-Shishtawy, Joel Höglund, Konstantin Popov, Nikos Parlavantzas, Vladimir Vlassov, and Per Brand

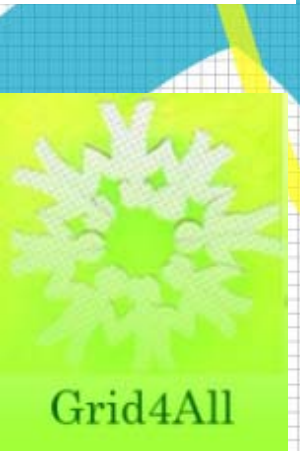
SELFMAN Workshop

21 October 2008, Isola di San Servolo (Venice), Italy

ahmadas@kth.se



ROYAL INSTITUTE
OF TECHNOLOGY



Outline

- Introduction
- Distributed Component Management System (DCMS)
- Yet Another Storage Service (YASS)
- Control Loop Patterns in YASS
- Conclusions
- Future Work

Introduction

- Grid4All
- Problem
 - Management by humans is expensive
 - More difficult for distributed applications
- Autonomic Computing & Self-*
- Single loop is not enough
- How to manage distributed applications using multiple control loops?

Outline

- Introduction
- **Distributed Component Management System (DCMS)**
- Yet Another Storage Service (YASS)
- Control Loop Patterns in YASS
- Conclusions
- Future Work

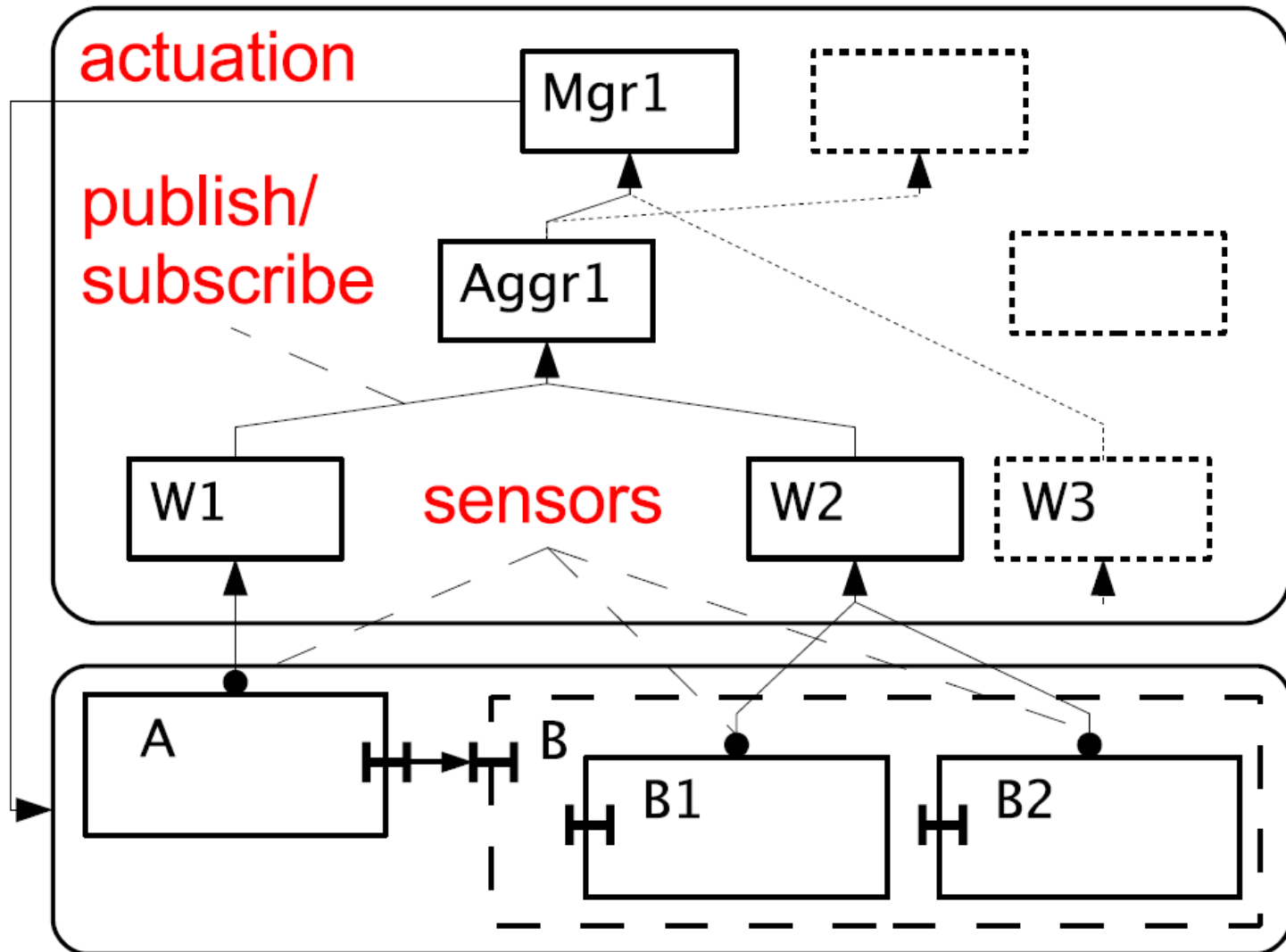
Distributed Component Management System (DCMS)

- Runtime system and model/API
- Runtime is a set of distributed containers
- Extends Fractal Component
- Separates functional and management parts (self-*)
- Management part is a network of Management Elements (MEs)

Distributed Component Management System (DCMS)

- MEs: distributed & Interact through events
- Used to construct Autonomic Managers
- MEs are divided into
 - Watchers
 - Aggregators
 - Managers
- Sensing/actuating is supported
 - Sensors
 - Actuation API (Deploy, Bind, Reconfigure, ...)

Application Architecture



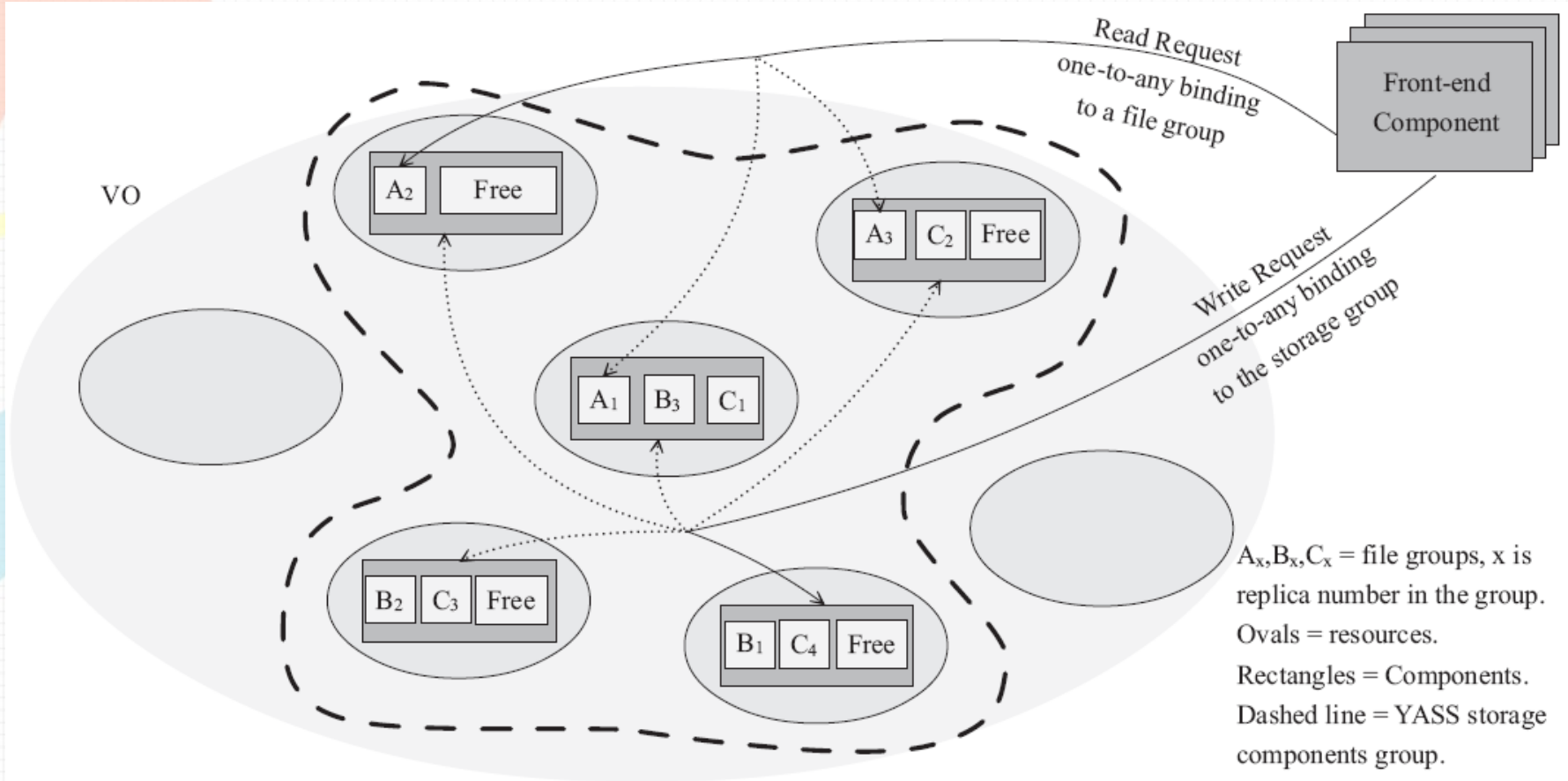
Outline

- Introduction
- Distributed Component Management System (DCMS)
- **Yet Another Storage Service (YASS)**
- Control Loop Patterns in YASS
- Conclusions
- Future Work

YASS

- Yet Another Storage Service
- Proof-of-concept distributed self-managing storage service built on DCMS
- Targets dynamic environments (resources join, leave, fail at any time)
- Used here to illustrate and discuss control loops

YASS Functional Part



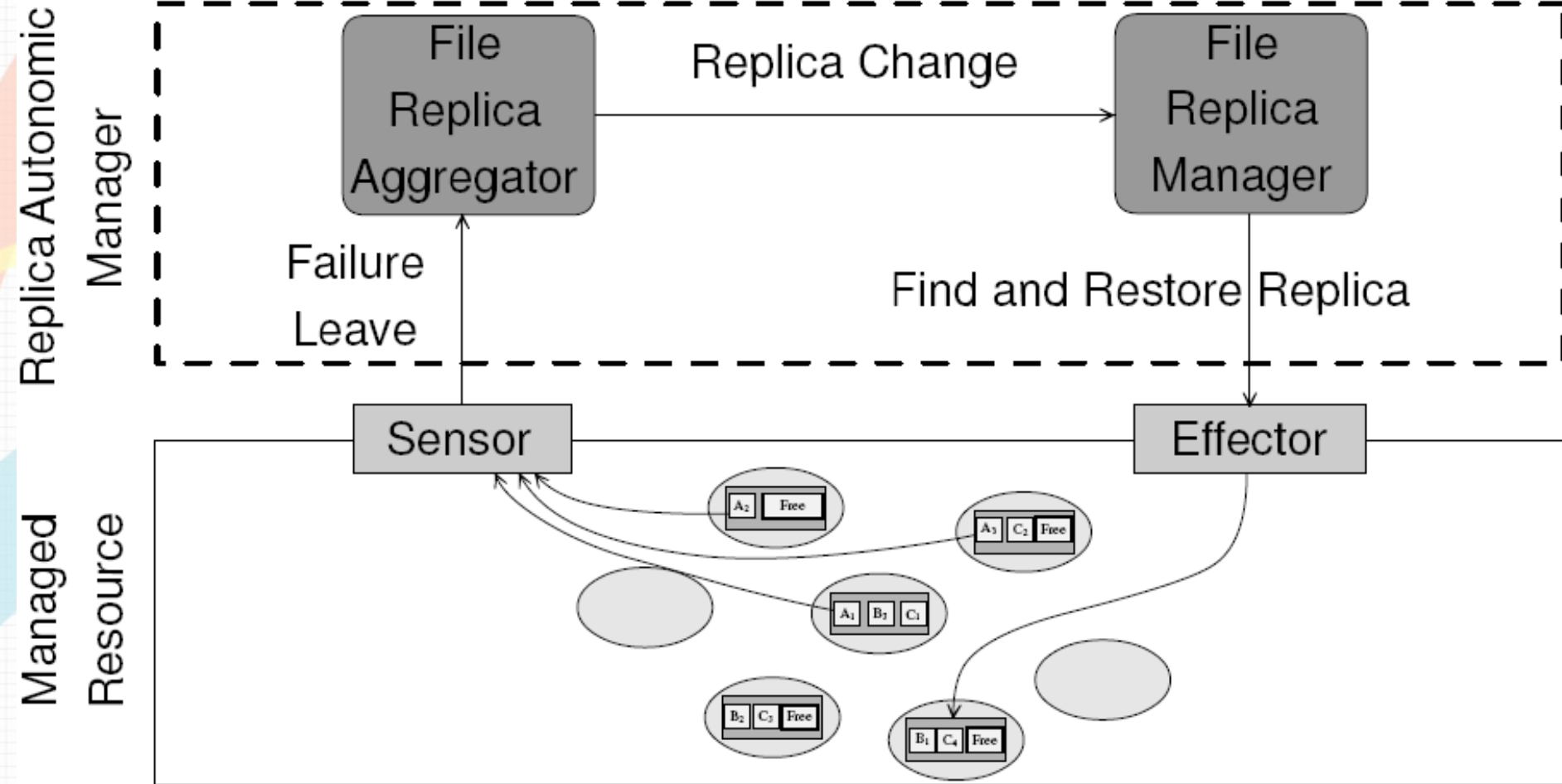
Outline

- Introduction
- Distributed Component Management System (DCMS)
- Yet Another Storage Service (YASS)
- **Control Loop Patterns in YASS**
- Conclusions
- Future Work

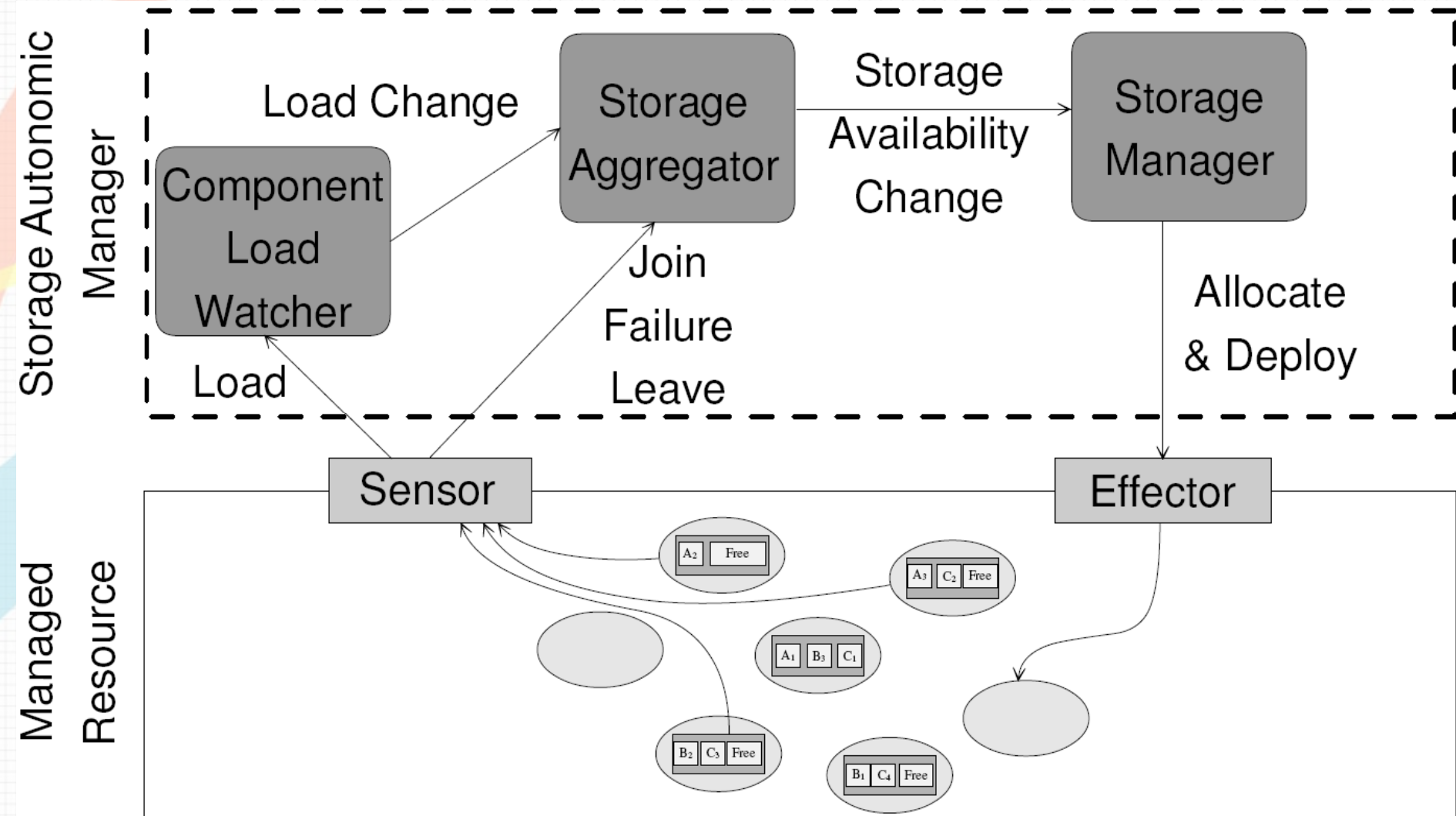
Basic Control Loops

- A control loop that does not coordinate with other loops
- Two in YASS:
 - Self-Healing: Maintaining replication degree of files
 - Self-Configuration: Adapt YASS to meet requirements under churn

Self-Healing Control Loop



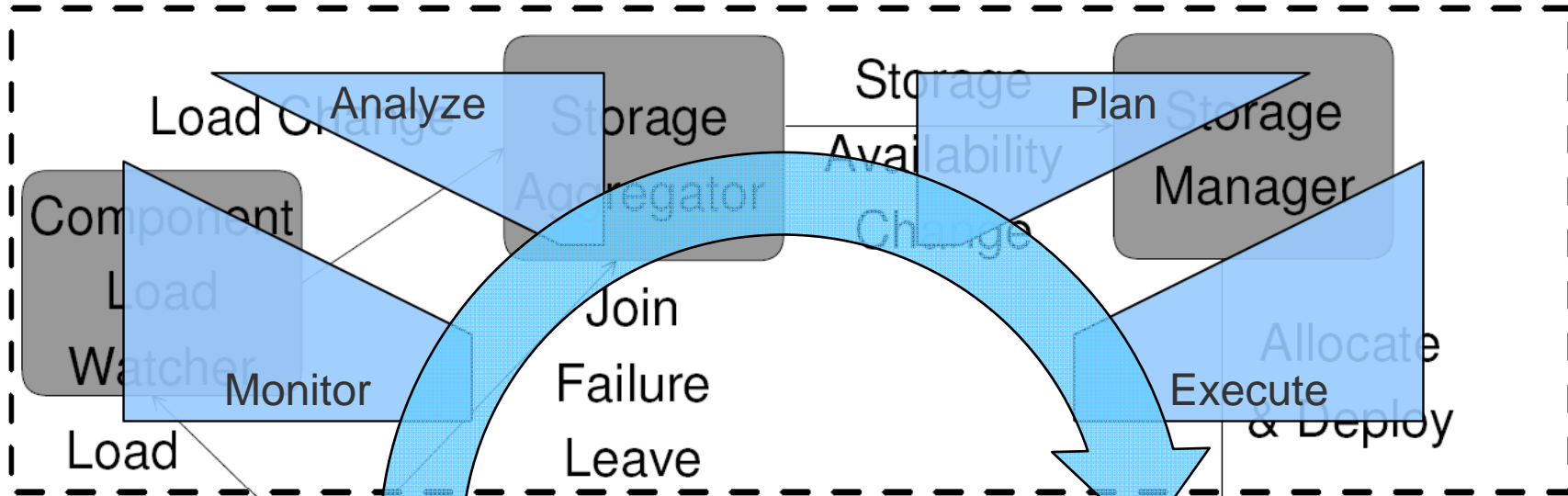
Self-Configuration Control Loop



Self-Configuration Control Loop

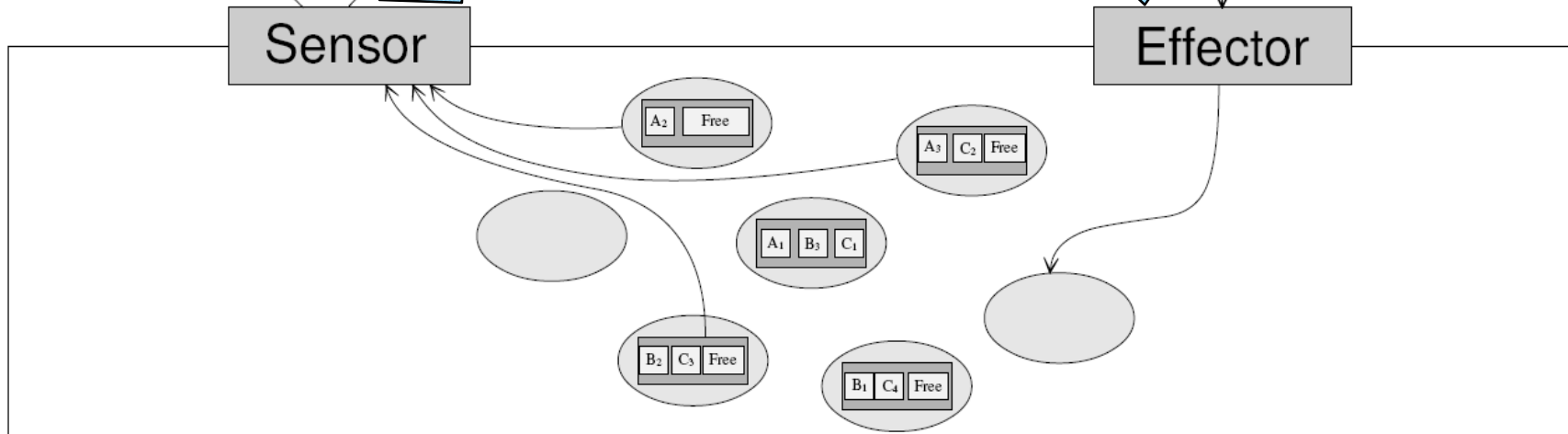
Storage Autonomic Manager

Manager



Managed Resource

Resource



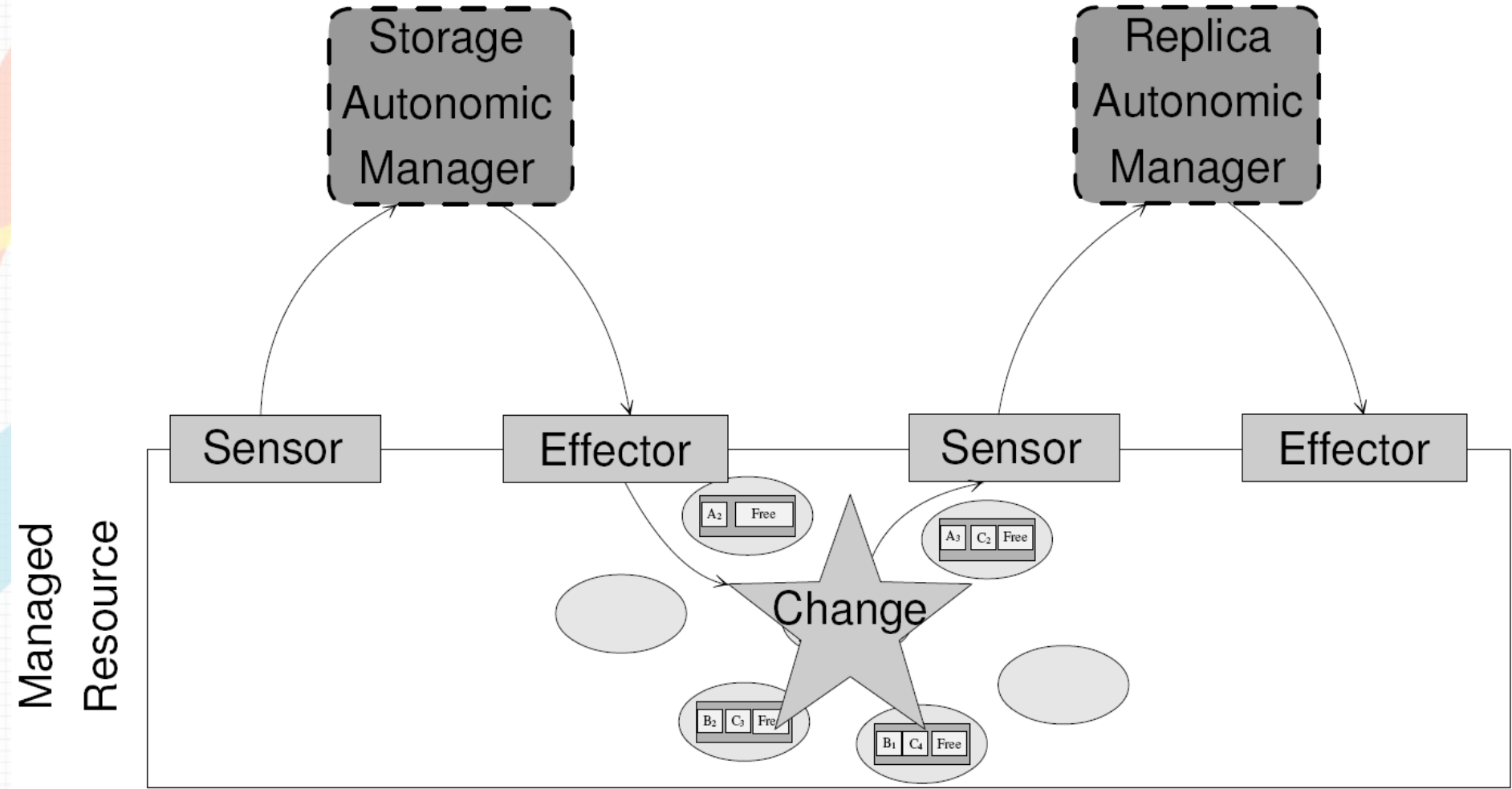
Example of Self-Management Code

```
public void eventHandler(Event e) {  
    StorageAvailabilityChangeEvent event = (StorageAvailabilityChangeEvent)e;  
  
    if (event.getTotalCapacity() < capacityLowThreshold) {  
        // find, allocate & add to group  
        ResourceId newResource =  
            myManagementInterface.getResource(preferenceHolder);  
  
        if (newResource != null) {  
            System.out.println("Found a new resource");  
            newResource = myManagementInterface.allocate(newResource);  
            ComponentId cid = myManagementInterface.deploy(newResource, depParams);  
            componentGroup.add(cid);  
        } else {  
            System.out.println("Cannot currently find a new resource");  
        }  
    }  
}
```

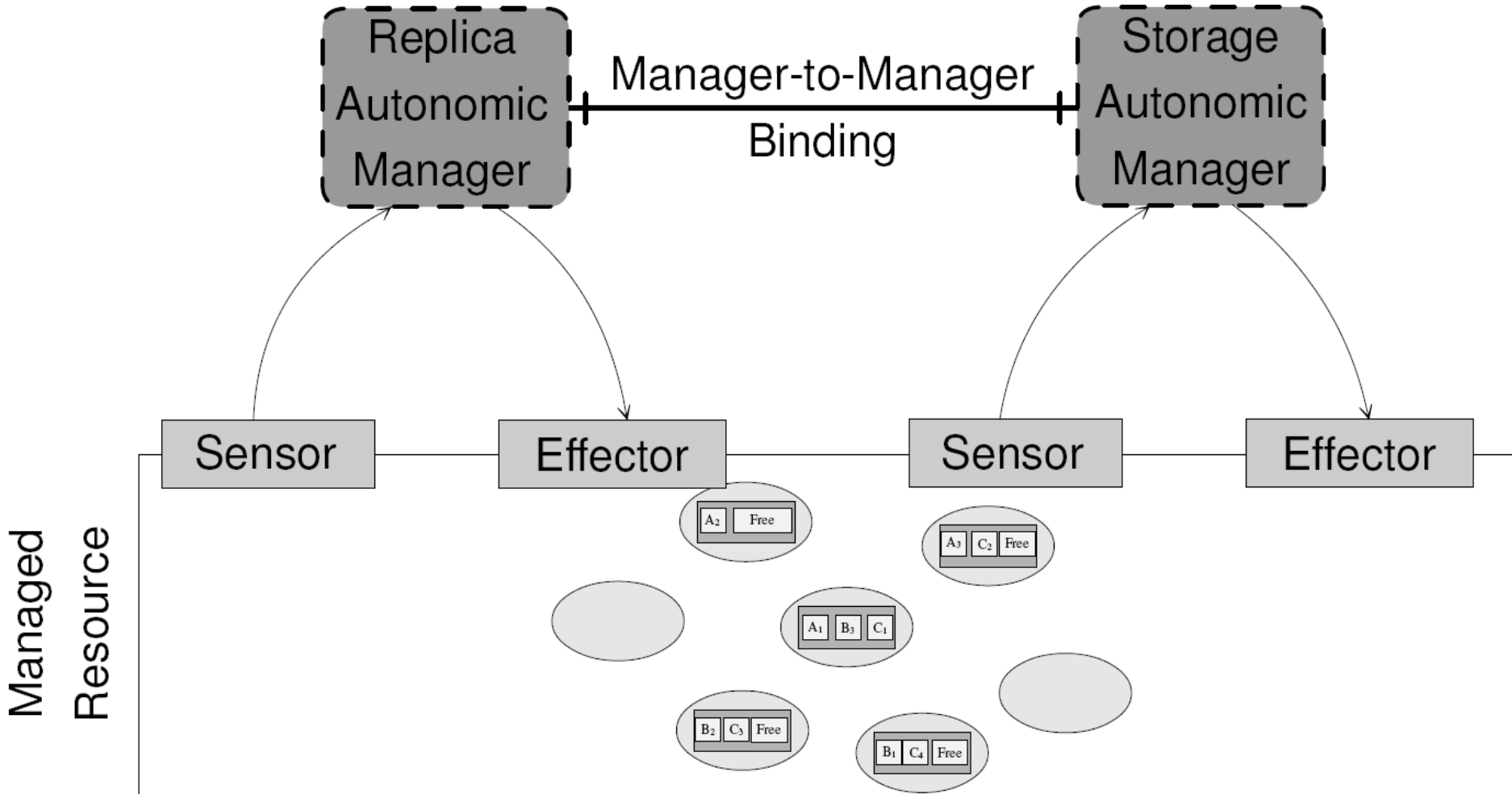

Coordinating Multiple Control Loops

- **Stigmergy**
 - Used for Self-optimization in YASS
- **Management-to-Management**
 - Used for coordinating basic loops
- **Hierarchical Management**
 - Used for self-optimization

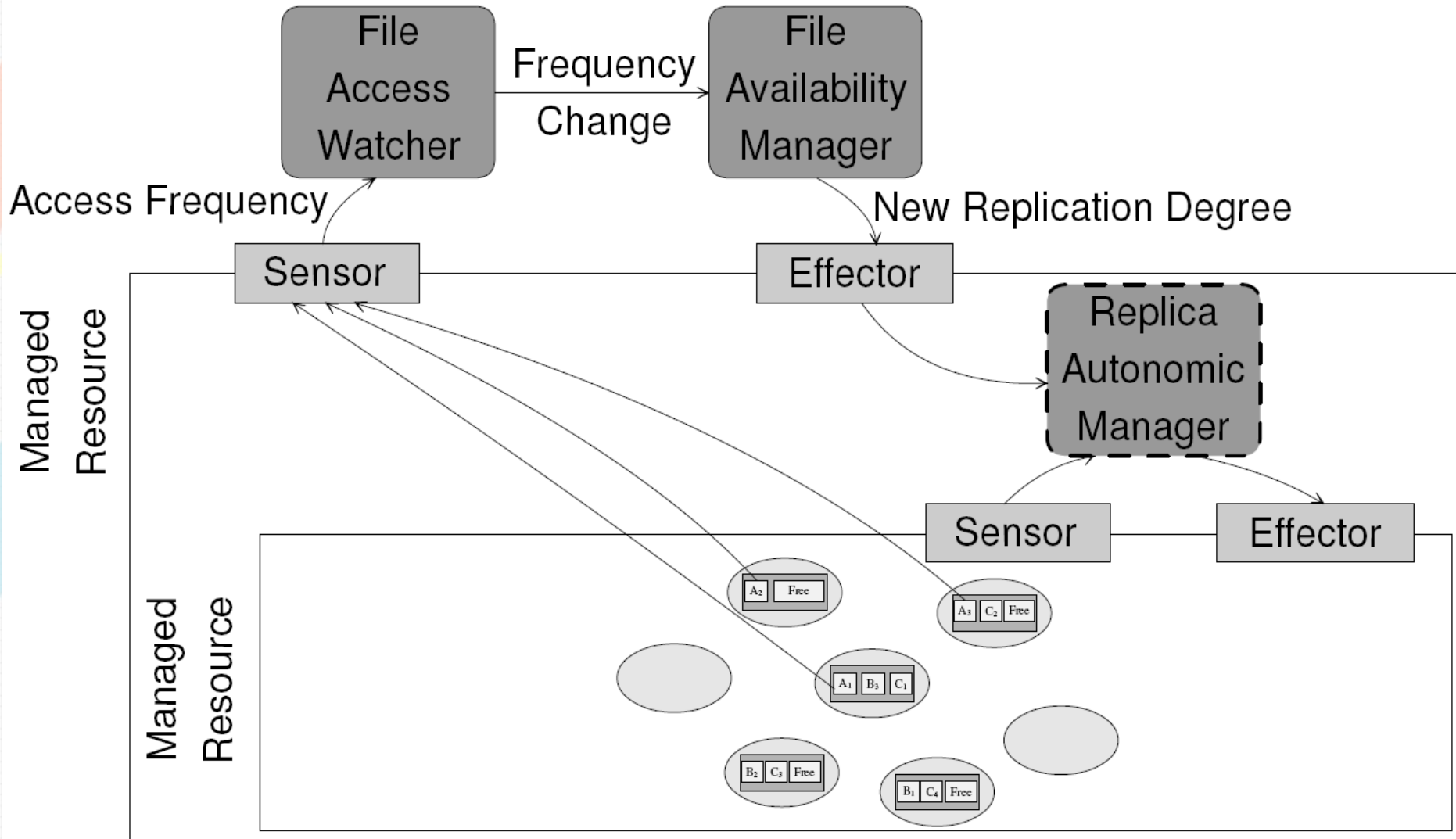
Stigmergy



Manager-to-Manager Interaction



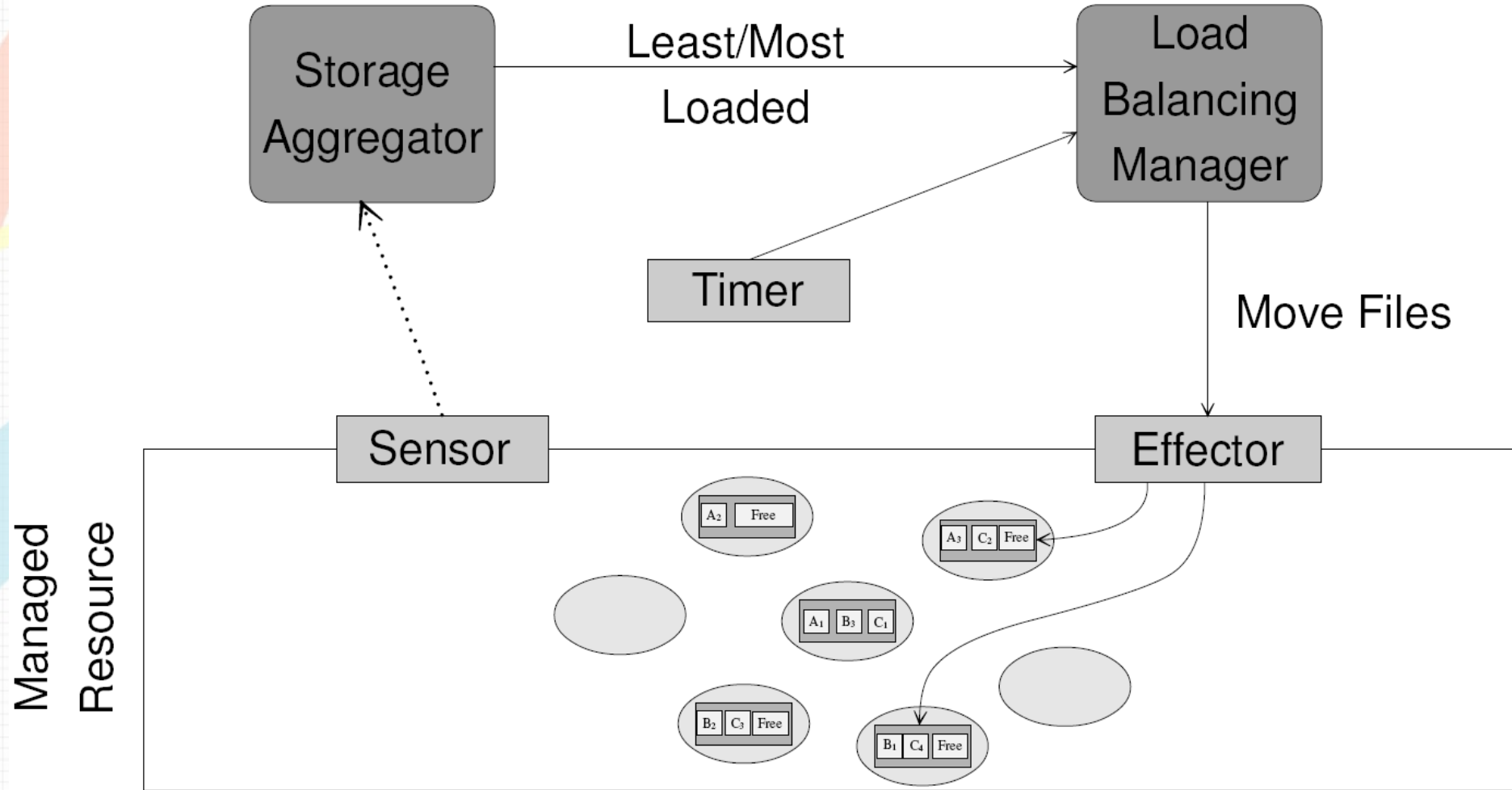
Hierarchical Management



Proactive Managers

- Implemented in DCMS using timers
- Example in YASS:
 - Load-Balancing control loop
 - Also shows sharing

Proactive Manager



Outline

- Introduction
- Distributed Component Management System (DCMS)
- Yet Another Storage Service (YASS)
- Control Loop Patterns in YASS
- **Conclusions**
- **Future Work**

Conclusions

- DCMS provides programming framework for constructing distributed control loops
- Distributed applications need multiple loops
- Multiple loops are usually independent but need to coordinate with other loops to improve efficiency
- Discussed three main interaction patterns
- Examples on how to apply them to YASS

Future Work

- Methods to analyze control loops
- Add more loops and study complex interactions and behaviors
- Formalize and generalize control loop patterns to be reusable specially for distributed applications.

Thank You :-)

Questions?

21 October 2008

Ahmad Al-Shishtawy

26



Backup Slides

21 October 2008

Ahmad Al-Shishtawy

27

Example of YASS deployment

