Self-Adapting Concurrency: The DMonA Architecture

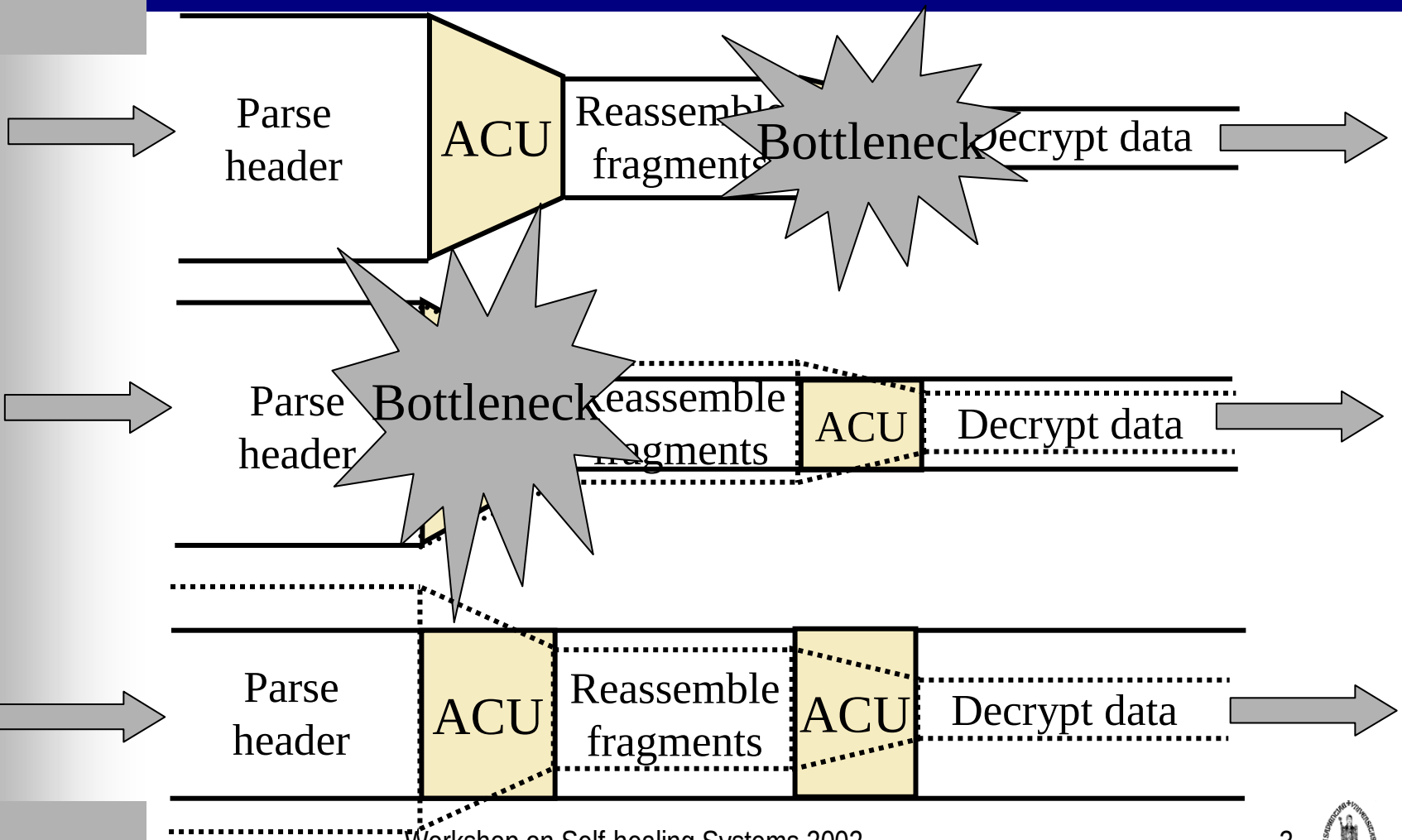
Sam Michiels Lieven Desmet
{Sam.Michiels,Lieven.Desmet}@cs.kuleuven.ac.be

Distrinet Research Group, K.U.Leuven, Belgium

Target applications

- Flow based internet services and underlying system software
  - Peak loads
    - § Slash-dot effect
    - § Not feasible to over-provision resources
  - Varying request types
    - § Static vs. dynamic web requests
    - § Varying security needs (types of encryption)

Target applications



New ideas/approaches

Component architecture (DiPS)

Pluggable self-healing architecture (DMonA)

Anonymous component interaction (DiPS)

Separation of concerns

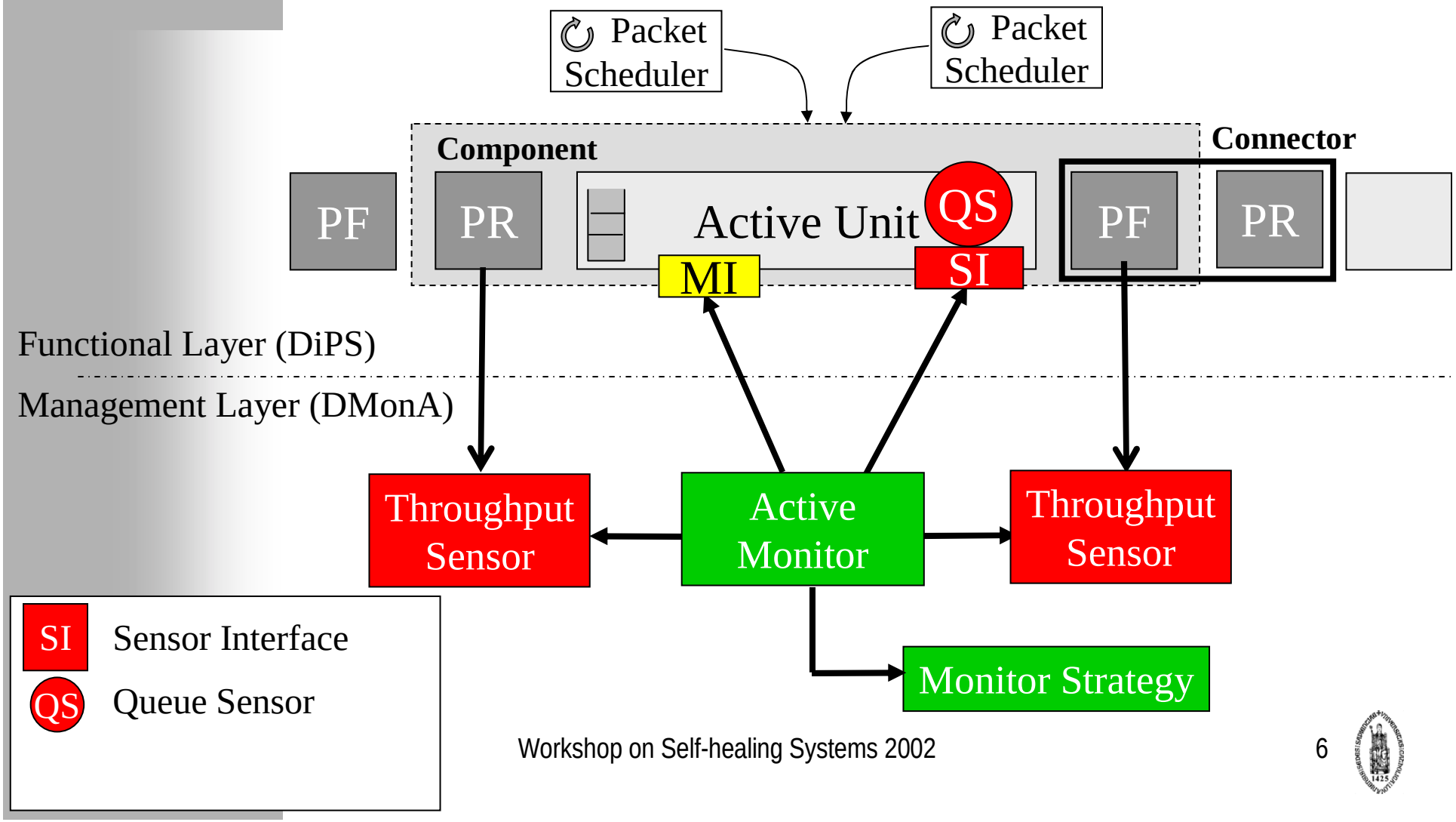
- Functional vs. concurrency units (DiPS)
- Functionality vs. management (DMonA)

Self-healing aspects of DMonA

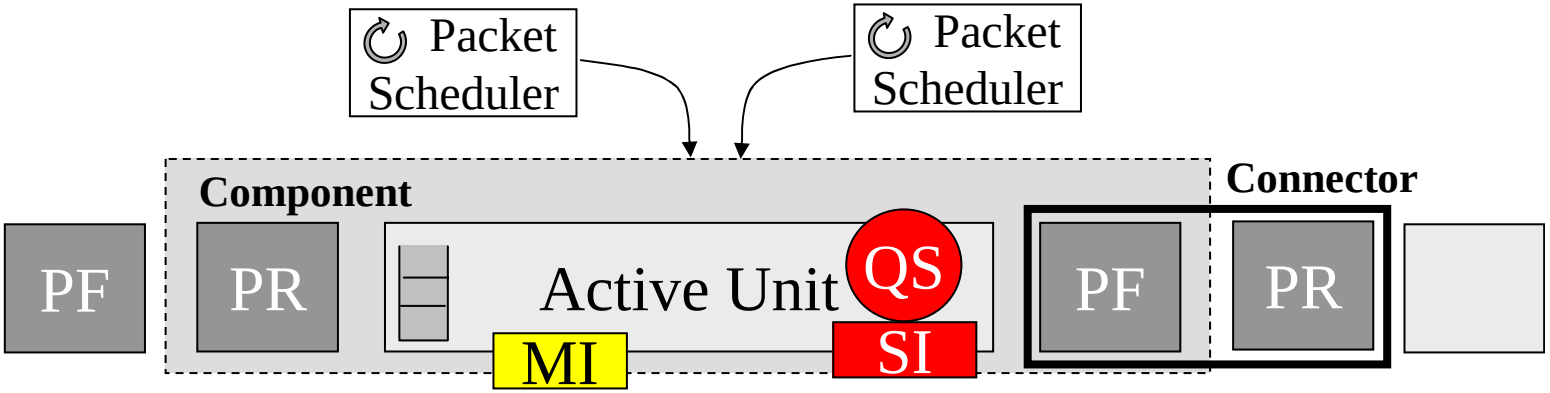
DMonA architecture

- Self-monitoring & interpreting
  - § Analysis & state sensors
- Prescribing appropriate measures
  - § Pluggable monitor strategies
- Self-healing
  - § Management interface
  - § Dynamic concurrency control for
    - improving performance
    - Graceful degradation

DMonA architecture



DMonA architecture



Functional Layer (DiPS)

Management Layer (DMonA)

KULeuven DistriNet Research Group

Conclusion

- Behavior adaptability (Concurrency)
- Separation of management and functionality
- Reuse of functional DiPS components
- DMonA is a runtime pluggable extension

Ongoing work

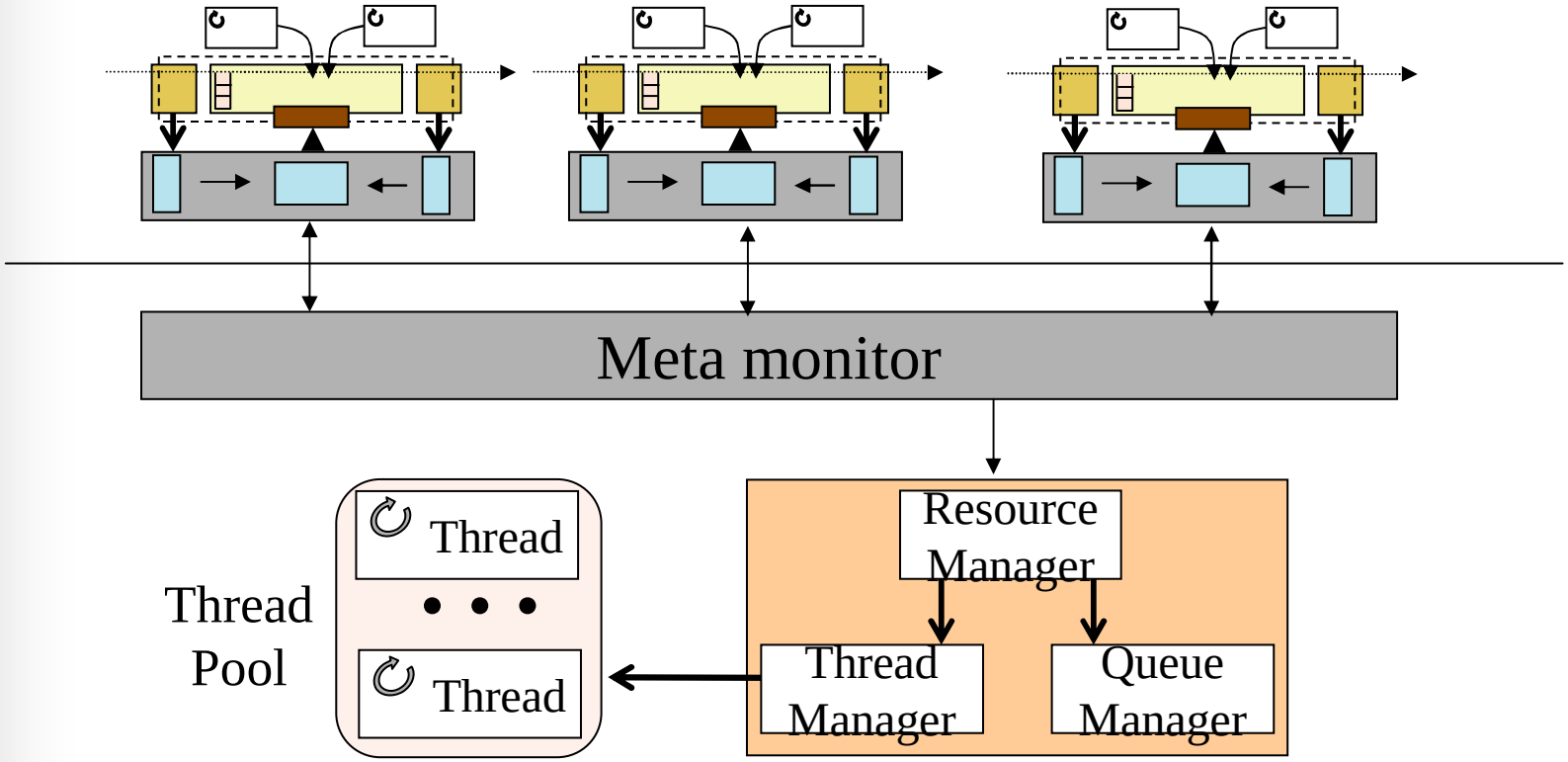
- Functional adaptability (Hot swapping)
- Combination of hot swapping extension with DMonA
- Environment sensors
  - Available system resources

Self-Adapting Concurrency: The DMonA Architecture

Sam Michiels Lieven Desmet
{Sam.Michiels,Lieven.Desmet}@cs.kuleuven.ac.be

Distrinet Research Group, K.U.Leuven, Belgium

Static & Dynamic Management



KULeuven DistriNet Research Group

Self-Adapting Concurrency: The DMonA Architecture