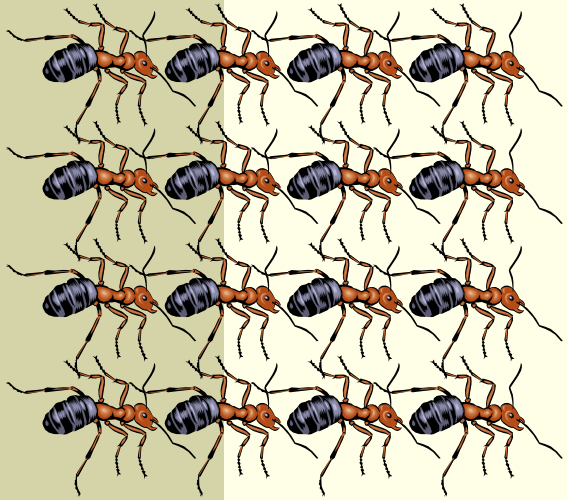
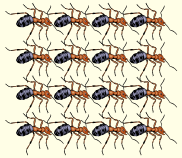
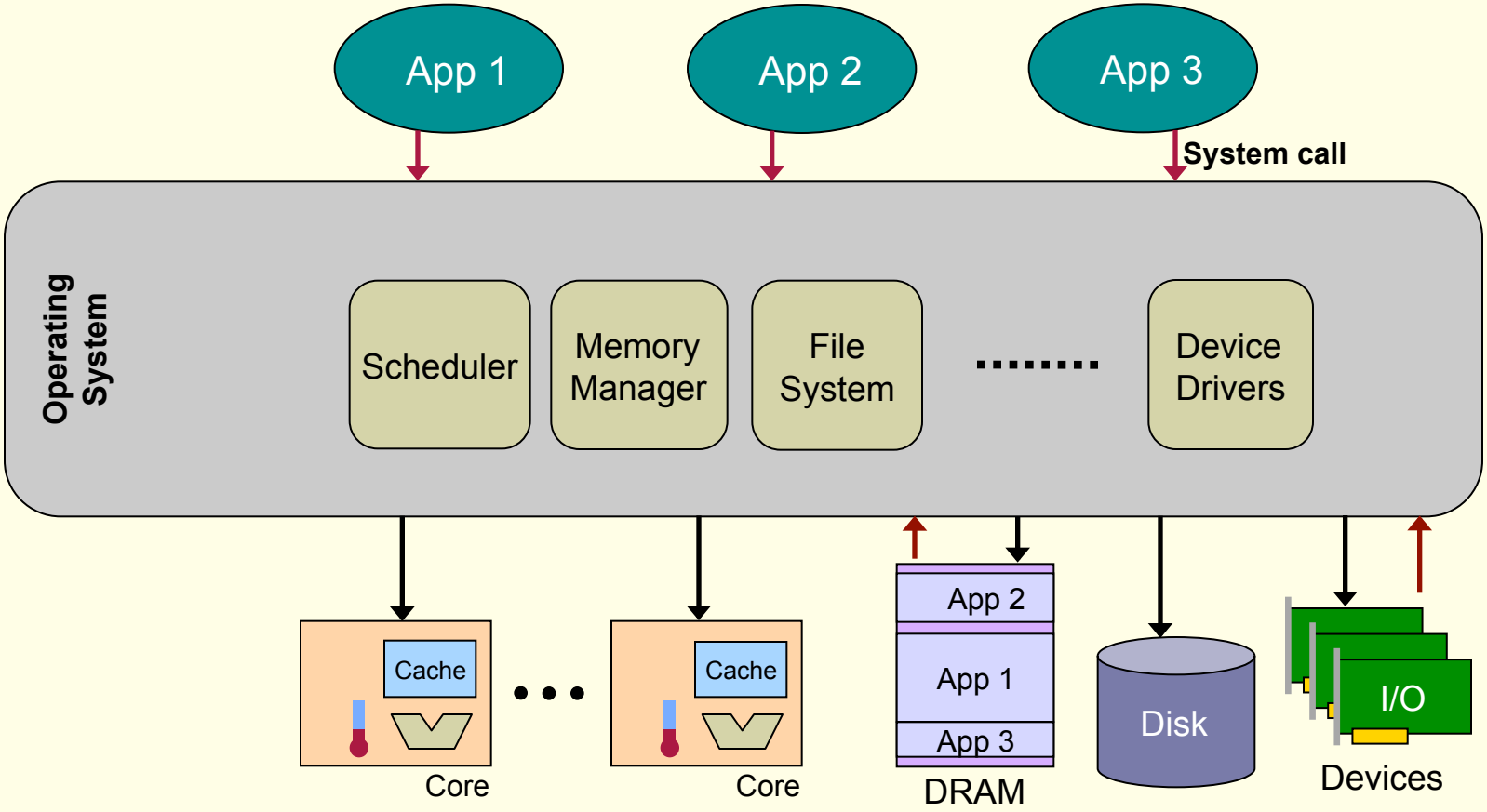


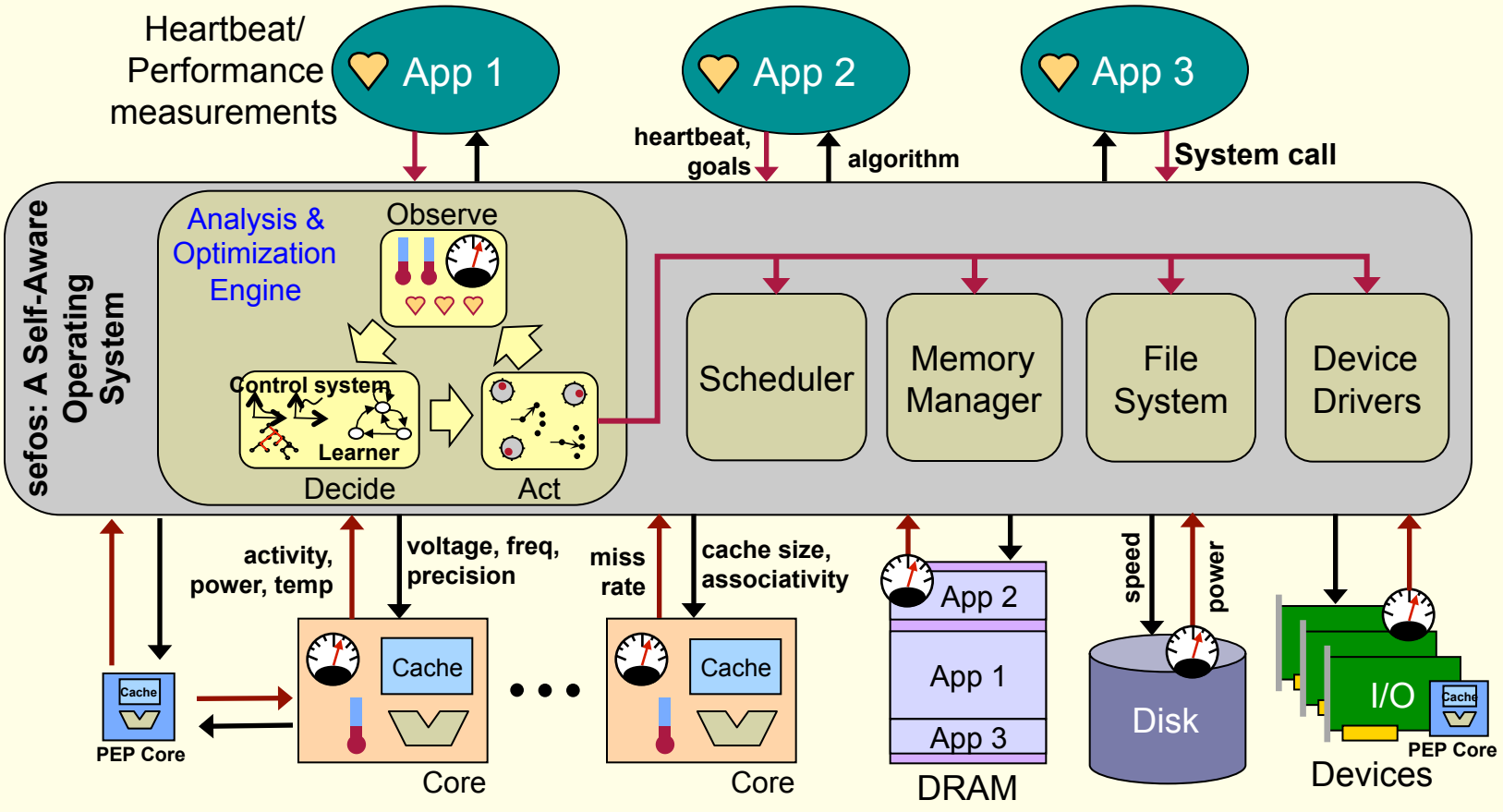
Sefos
A self-aware factored operating system



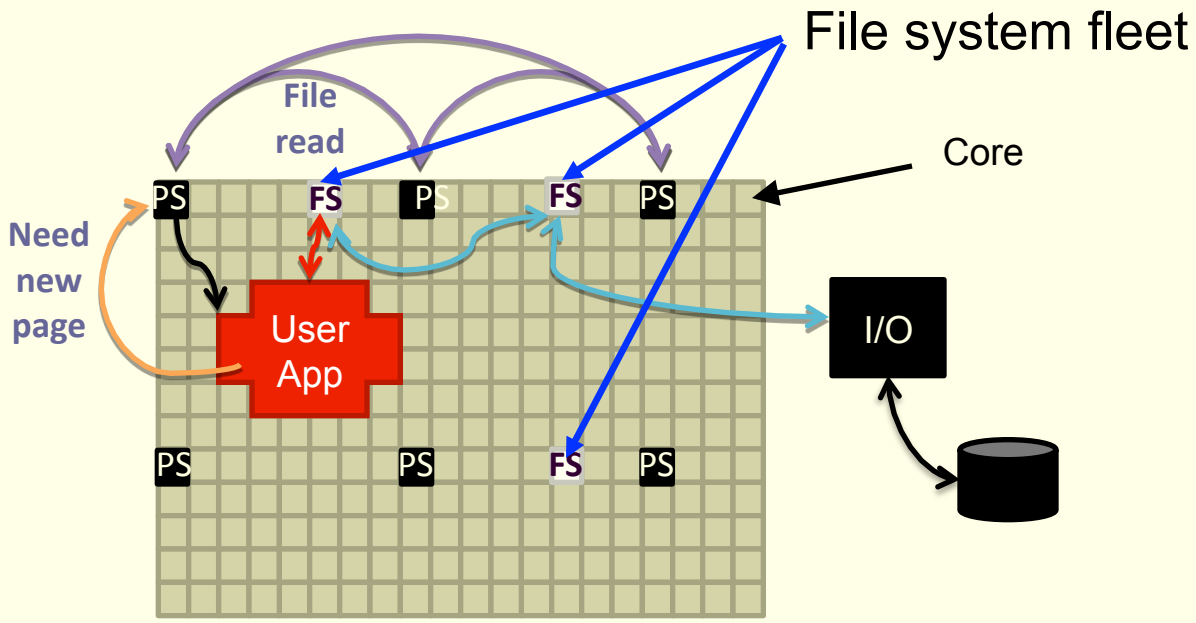
A Traditional OS



Sefos: A Self-Aware OS



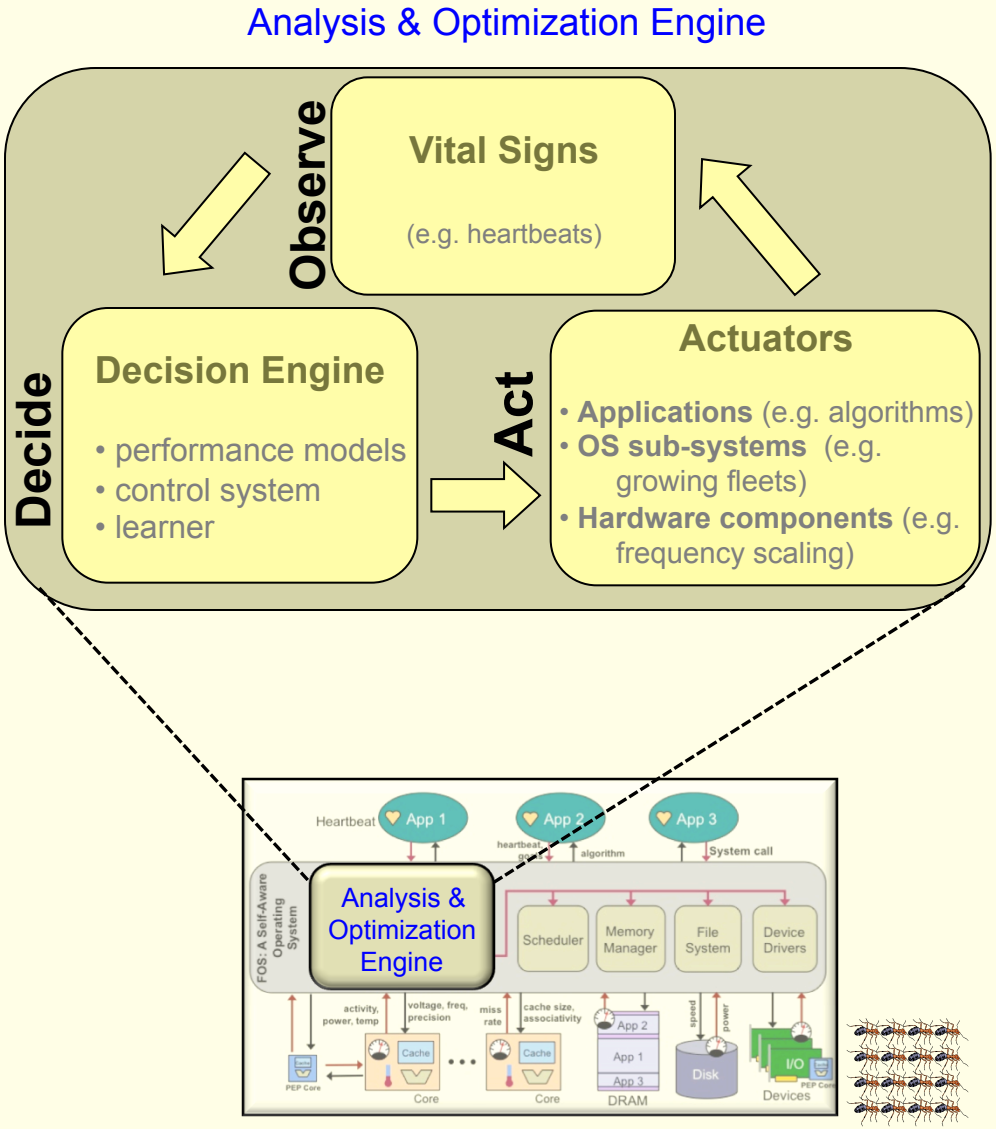
Sefos is Factored



- OS factored into services (e.g. name service, page service PS, file service FS)
- Each service is further factored into a ***fleet*** of distributed servers
- Each server is bound to a core
- Application cores message a particular server core to utilize service
- Server cores collaborate/communicate to implement needed OS service

Sefos is Self-Aware

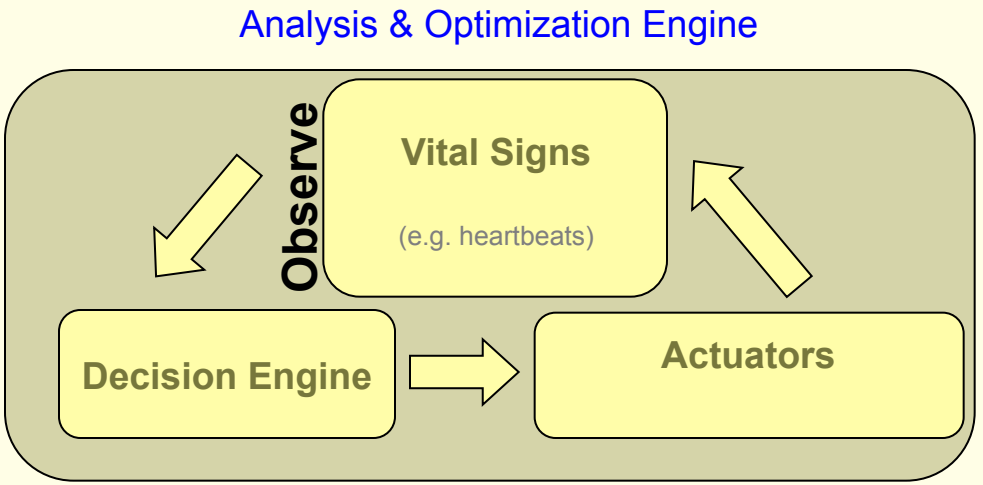
- Sefos Analysis and Optimization Engine
  - A closed feedback loop of ODA
  - Each component provides a new OS service:
    - The observer provides vital signs services
    - The decision engine provides performance models and AI learning engine
    - The actuators modifies system status
  - Each component is either implemented as a fos fleet or integrated into another S/W components.



Sefos: Building self-awareness into fos

Vital Signs Fleet

- Observes the status of software and hardware components.
- Provides a distributed, global knowledge-base of the system
- Implemented as a fleet, storing the information in a distributed data object
  - key-value store



- Collects measurements from:

1) Applications

e.g. Apps heartbeats, Application-specific measurements (fps in a video encoder or flops in a scientific app)

2) OS subsystems

e.g. Utilization ratio of the file system fleet

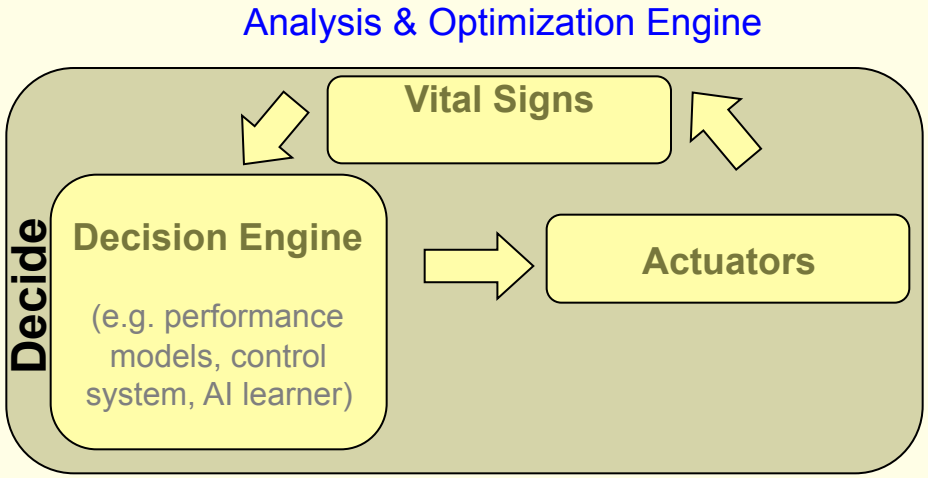
3) Hardware components

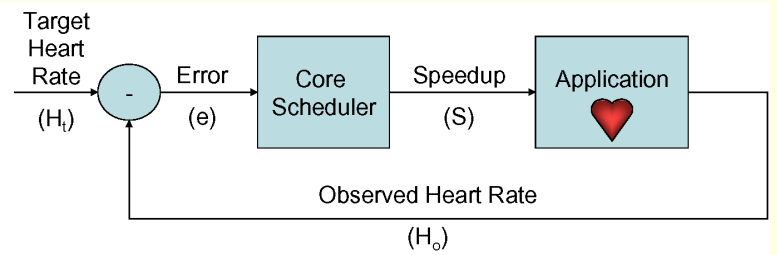
e.g. temperature, core frequency, power, cache miss rate

Sefos: Building self-awareness into fos

Decision Engine

- A new OS service
- Implemented through a fleet of servers for scalability
- Process input from the vital signs service
- Provides two approaches to runtime decision making:
  - Machine learning
  - Classical control theory

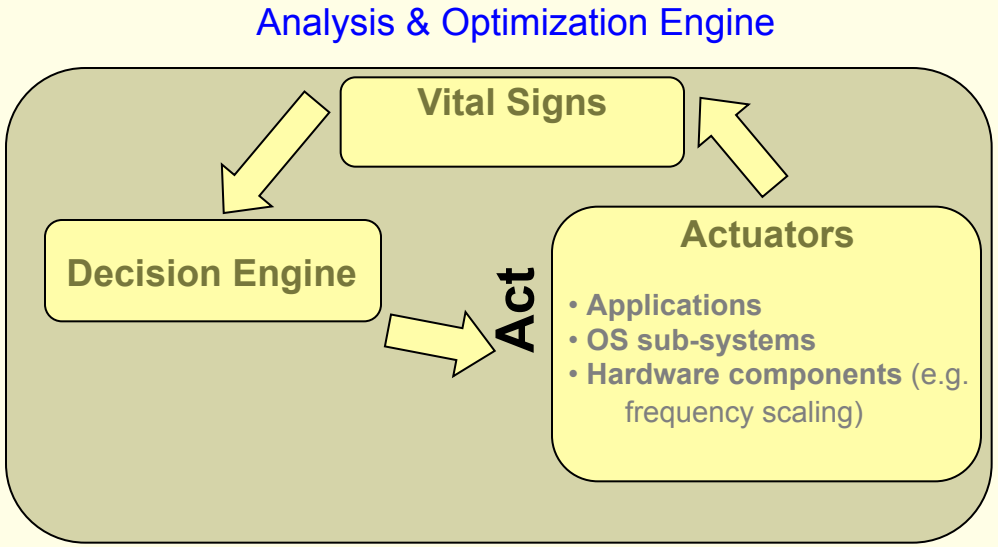




Sefos: Building self-awareness into fos

Actuators

- Actions allowing the system to adapt at runtime
- Three sets of actions, based on their impact:
  - Application actions
    - Allocating or de-allocating cores to an application
    - Switching between algorithms
    - Migrating processes for better data locality and cache usage
  - OS services actions
    - Growing and shrinking fleets
    - Migrating servers
  - Hardware actions
    - Frequency scaling to save power



- Implemented as extension to fos fleets, OS tools for hardware actions or application-specific actions

Observe-Decide-Act Loop in Sefos