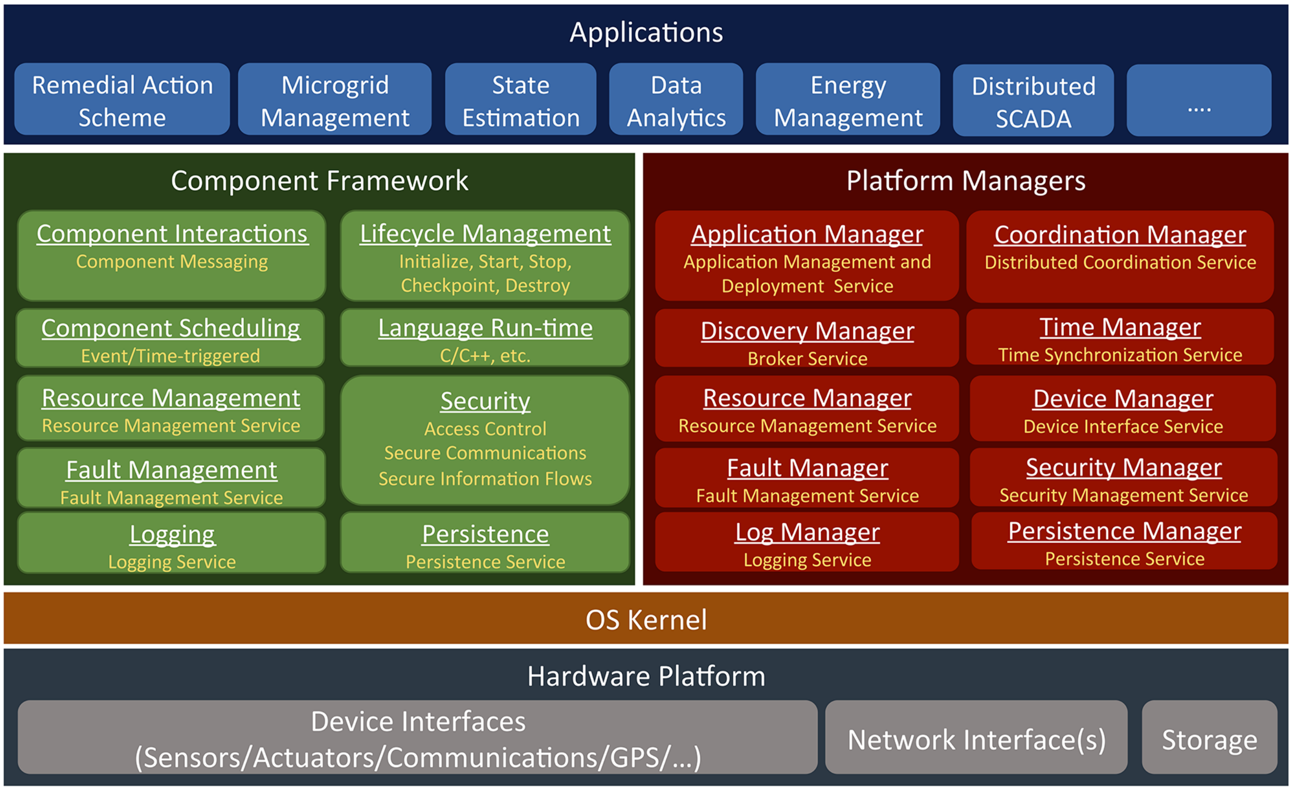
RIAPS Architecture

Architecture



Each actor encapsulates run-time layers of RIAPS that provide

- component framework that defines a concurrent model of computation for building distributed applications
- resource management framework for controlling the use of computational resources
- fault management framework for detecting and mitigating faults in all layers of the system
- security framework to protect the confidentiality, integrity, and availability of a system under cyber attacks
- fault tolerant time synchronization service
- coordination framework for coordinated computations and actions across the network
- application's business logic can be kept separated from the low-level details the framework

Infrastructure

blocked URL
The RIAPS infrastructure features

- multi-tenant computing nodes hosting decentralized applications firewalled from each other on a communication network
- scalable deployment and management framework for the administration and control of distributed applications from a control room
- discovery framework for establishing the network of interacting actors of an application
- messaging framework for facilitating interactions among actors
- coordinated time synchronized scheduled action through consensus with logical dynamic grouping of nodes

Development Tools

blocked URL
Tools are provided in the form of a model-driven development environment (MDE) allowing

- increased application developer productivity
- migration of accidental complexity during development
- use of domain-specific modeling language for compact declarative specification of software components and the composition of applications
- developer to focus on solving power system problems while low-level software details are handled by the tools

Time Synchronization

blocked URL

For precisely timed measurements and operations the applications need to be aware of and monitor the delays introduced by the network and the software layers.
Use cases:

- synchronized distributed action (e.g. breaker activation)
- detecting network bottlenecks or Denial of Service attacks
- fallback when global time synchronization is not available
- communication profiling and tracing

- Distributed Coordination
  blocked URL
  Use cases:
  - Group Membership - components of one or more applications form groups during operation for message sharing
  - Leader Election - a single component becomes designated as an organizer of tasks (or decision maker) among several distributed components
  - Time-synchronized Coordinated Action - coordinated agreement amongst distributed nodes regarding when a time-synchronized action should be performed

Fault Tolerance

blocked URL
Detected Faults:

1. reported application process termination
2. unreported application process termination
3. application resource limit violation
4. application component operation deadline violation
5. unexpected service termination
6. operating system crash
7. network link failure
8. network node failure
9. application deployment failure
10. loss of connectivity to control station

Using Detection/Isolation/Recovery Paradigm:

- Detection - recognition of an anomalous situation
- Isolation - finding the root cause of the problem
- Mitigation - action taken to mitigate the effect of faults (handled by application developers)

Security

blocked URL
Protects against security threats by ensuring

- confidentiality and integrity of communications by encrypting all network communications and ensuring that messages were not tampered with
- availability of resources by providing facilities for strict access control to resources and moderating processing activities to mitigate DDoS attacks
- confidentiality of data by ensuring strict access control of data owned by an application to protect against malicious or faulty application code
- applications will be remotely deployed and controlled through the use of cryptographic signatures on the application binaries to be installed