A RESTful Approach to the Management of Cloud Infrastructure

Swit Phuvipadawat
Murata Laboratory

“A RESTful Approach to the Management of Cloud Infrastructure”

- Hyuck Han, Shingyu Kim, Hyunsoo Jung, et.al
- 2009 IEEE International Conference on Cloud computing

Outline

- Introduction
- REST and Its Application
- CMS (Cloud Management System) Architecture
    - Design of the REST-based Manager
    - Design of the REST-based Agent
    - Management Information Model
- Conclusion

Introduction

- The concept of **REpresentational State Transfer (REST)** was introduced in 2000 by Roy Fielding.
  - Resources are identified by Uniform Resource Identifiers (URI).
  - Resources are manipulated through their representations.
  - Messages are self-descriptive and stateless.

Introduction

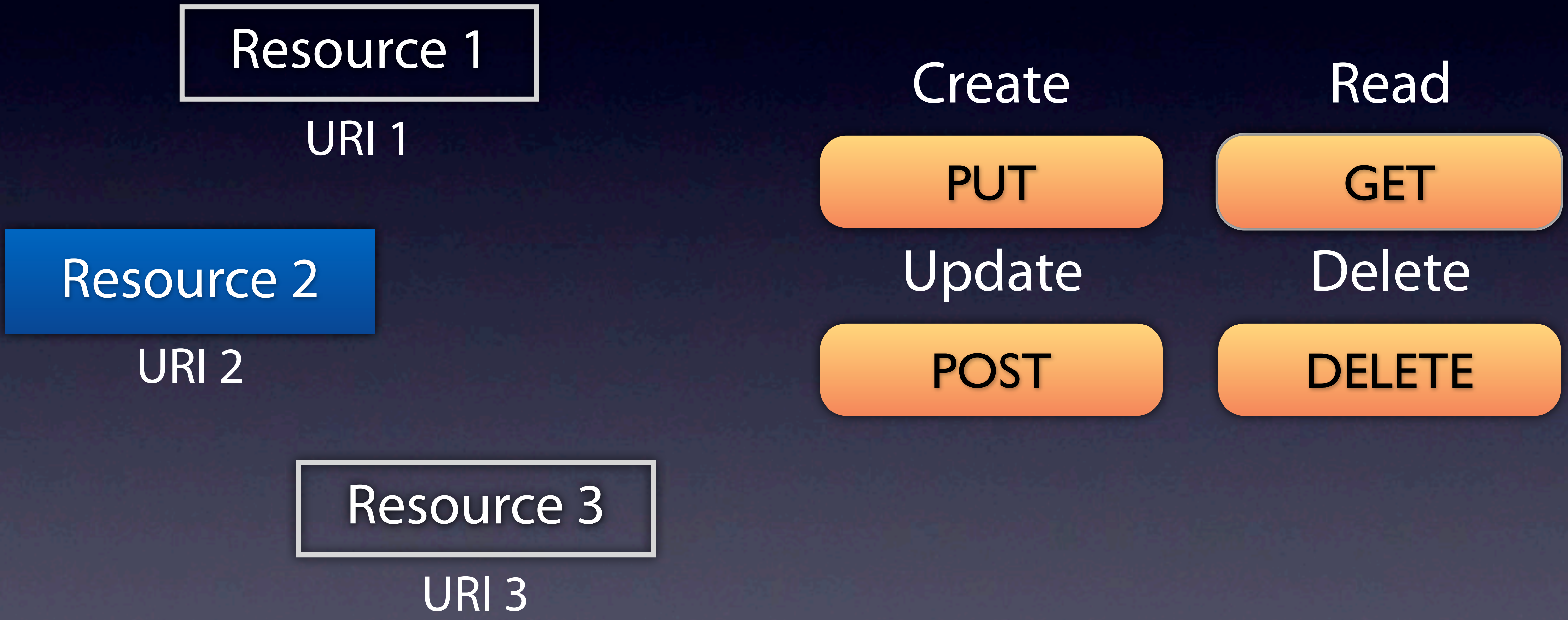
- Many frameworks have been implemented according to REST approach.
  - *E.g. Java Restlet, Rest in Python (RIP), Microsoft's Azure platform, etc.*
- However REST is not yet fully taken into the management systems for cloud infrastructure.
- The authors showed that RESTful approach can be useful for building such infrastructure.

RESTful Cloud Management System

- The authors of this paper stated the following contributions:
  - ✓ RESTful Web services can replace software components of existing management systems.
  - ✓ CMS architecture is proposed.

REST and Its Application

REST in Action



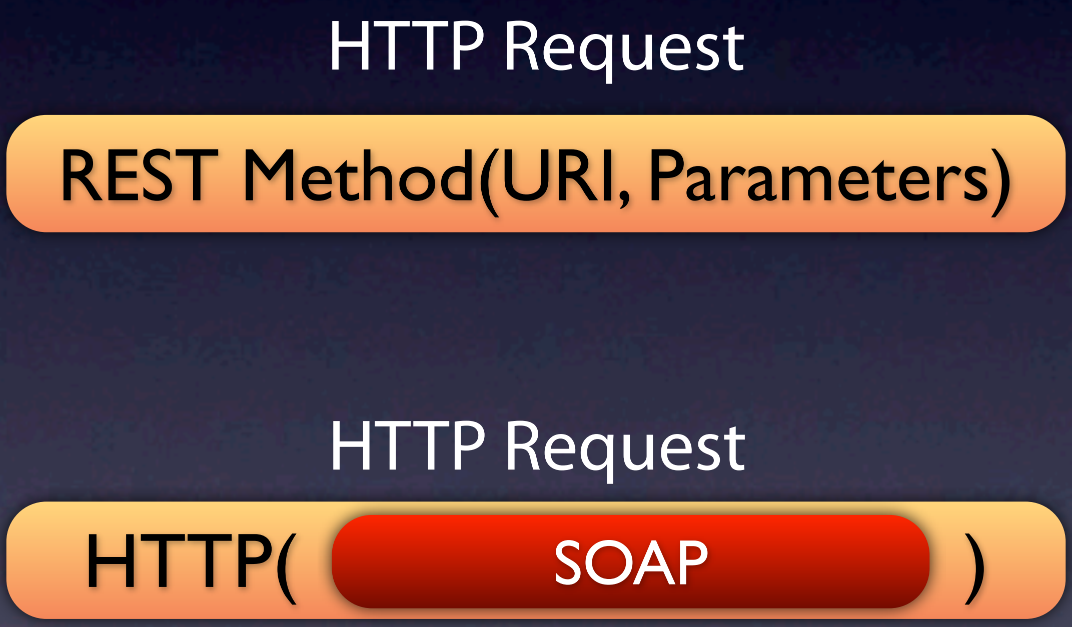
Each resources are identified by URI

Resources are manipulated by a set of verbs

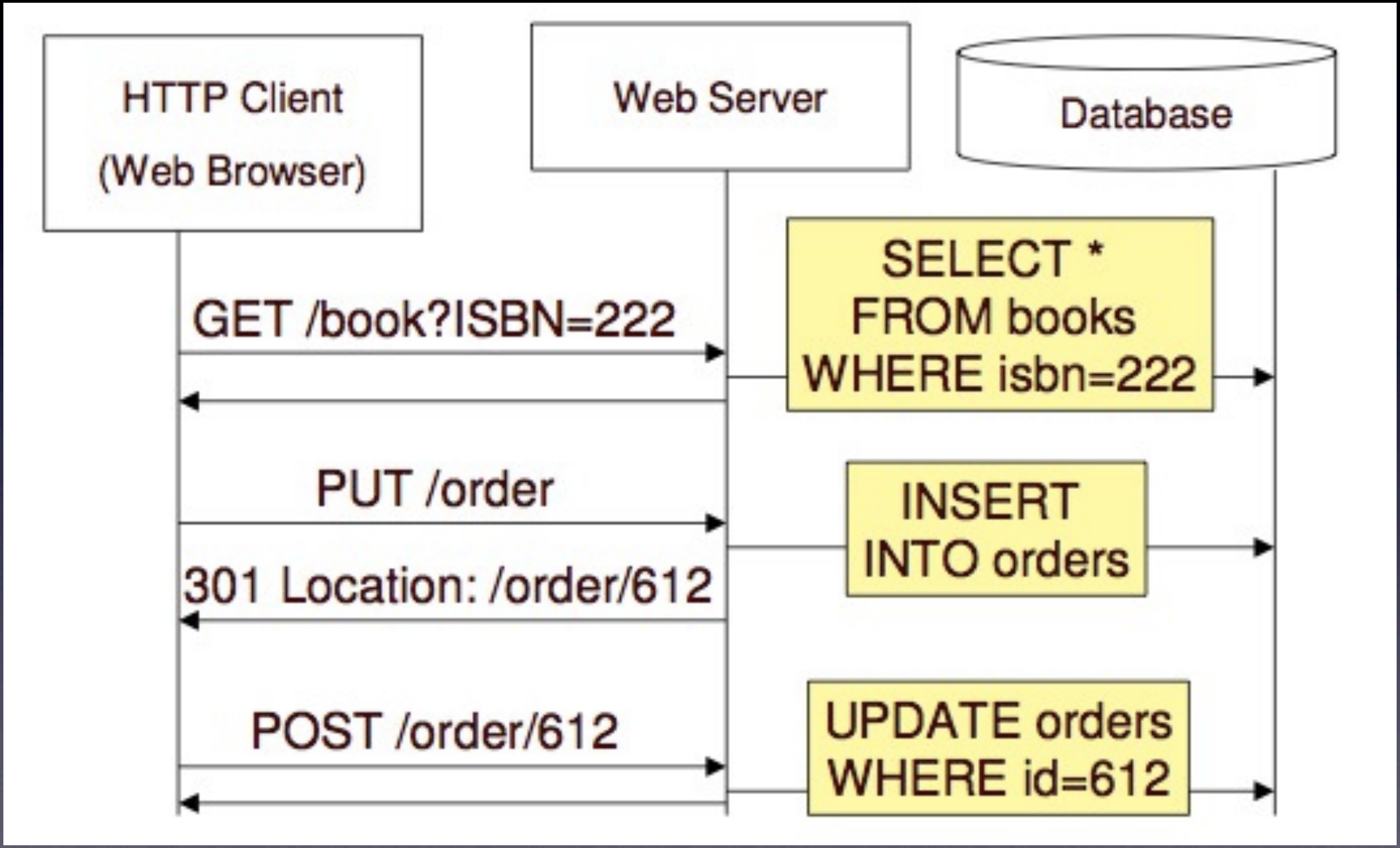
REST and Its Application

Besides REST there’s SOAP

- REST is used to describe simple interfaces over HTTP without additional messaging layer such as *Simple Object Access Protocol (SOAP)*.
- People often debate over issues between **REST** and **SOAP**.

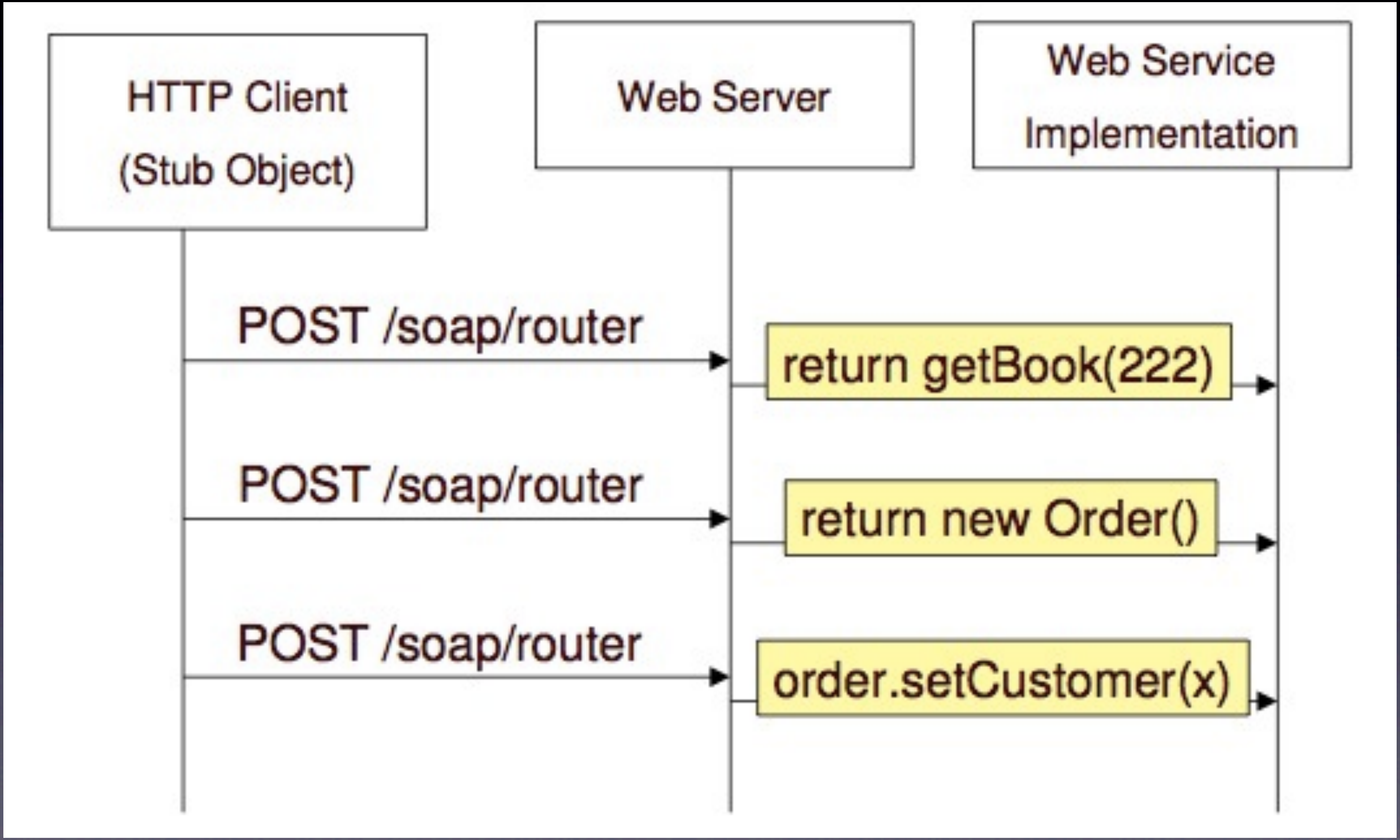


RESTful Web App Example



http://www.systems.ethz.ch/education/courses/fs09/web-services-and-service-oriented-architectures/slides/SOAwithREST.pdf

SOAP Web Service Example



Debate

SOAP

Strengths

- Protocol transparency
- Independent of the underlying protocol
- Rigid, type checked

Weakness

- Need a proper design to avoid leakage across abstraction levels
- Interoperability problems
- Complex

REST

Strengths

- Simple, lightweight
- Follows W3C/IETF standards
- URI allows resources to be discovered easily
- Scales well with caching

Weakness

- Some system only accepts GET and POST
- For GET requests, the size must be less than 4KB

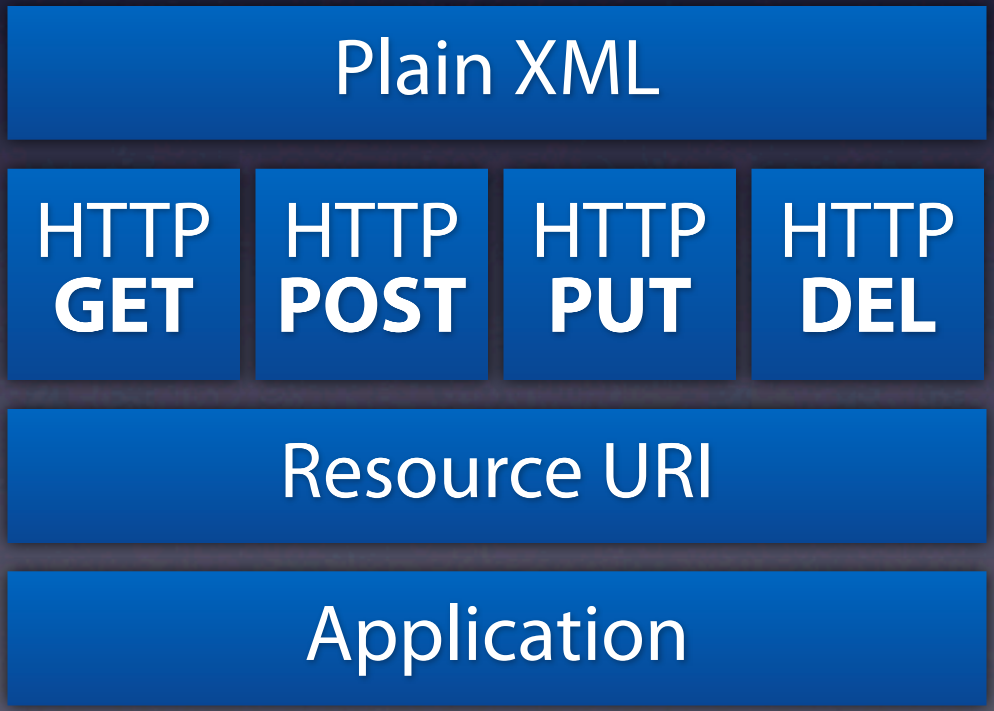
Main difference: REST vs SOAP

"The Web is the universe of globally accessible information" —Tim Berners Lee

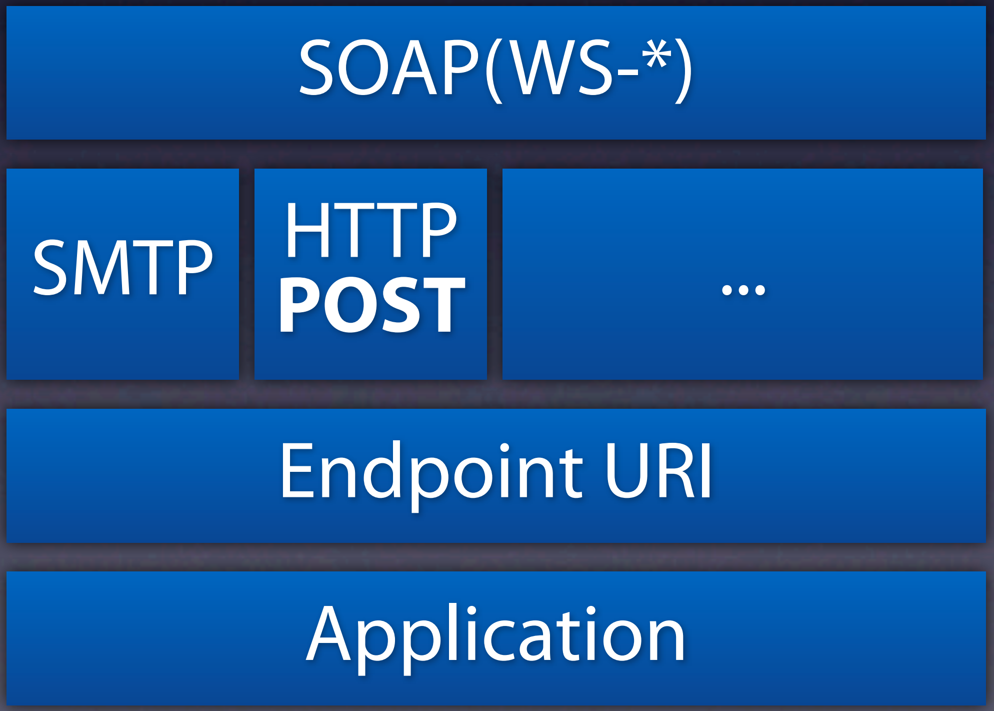
"The Web is the universal transport for messages"

Applications should publish their data on the Web (through URI)

Applications get a chance to interact but they remain "outside of the Web"



REST



SOAP

REST and Its Application

- REST is widely used because of its advantages.
- McFaddin et al. proposed RESTful data service for mobile environment.
- Volkel proposed a RESTful wiki architecture.
- Amazon S3, Yahoo, Twitter etc. provide REST API for their users.

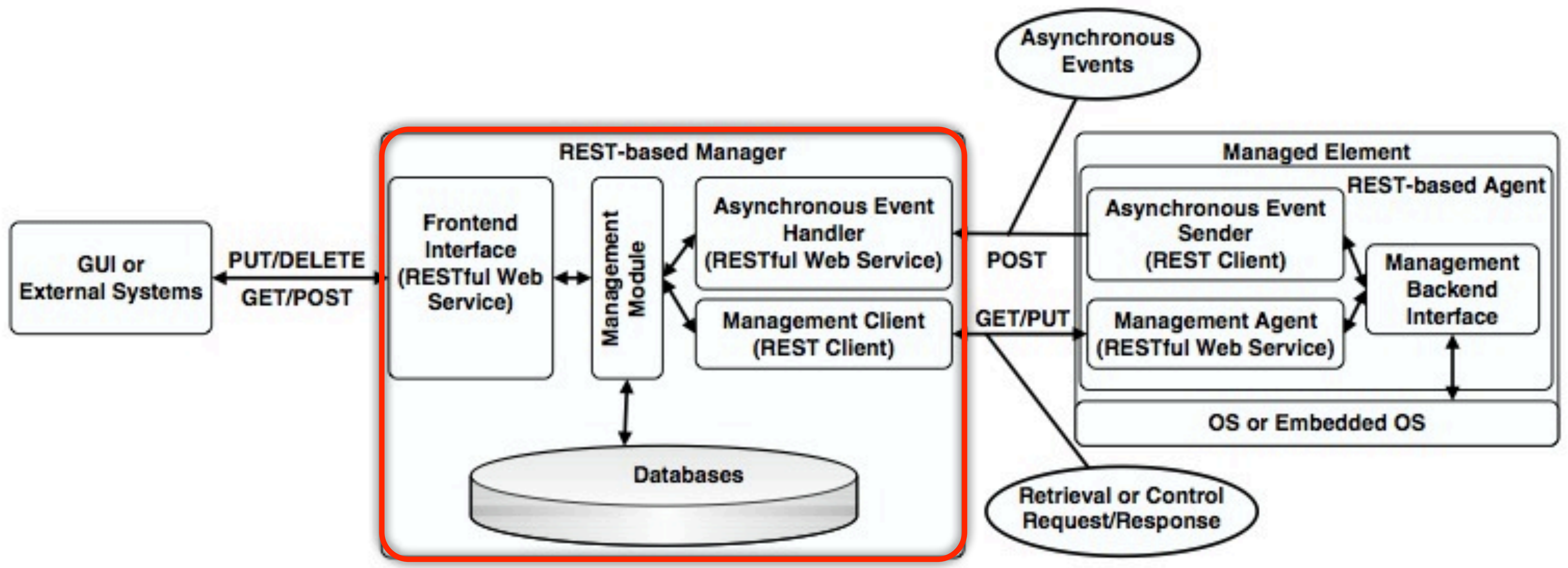
Management Systems

- A management system is composed of three components: *GUI manager*, *manager* and *managed elements (agents)*.
- Manger uses a centralized database or a federation of databases to manipulate management information.
- Many protocols such as CORBA, SOAP, XML-RPC, Enterprise JavaBeans, HTTP and user-defined protocols are used between external systems and manager.
- Sharing information among management system is difficult because there are a variety of management information model and protocols.

Management Systems

- Replacing those protocols with with REST would ease the difficulty of sharing information.
- REST also allows management systems to be easily decentralized.
  - ➡ Management information can be modeled as a resource identified by URI.

CMS Architecture



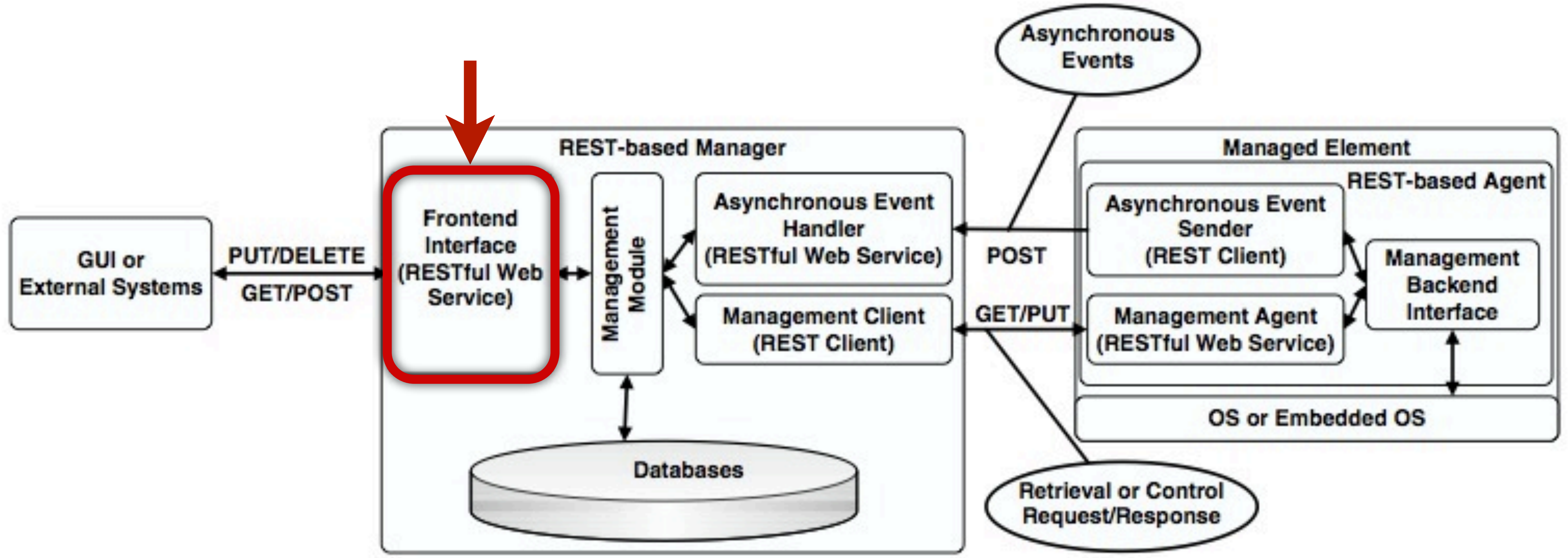
Asynchronous Events
REST-based Manager
Managed Element
REST-based Agent
GUI or External Systems
PUT/DELETE
GET/POST
Frontend Interface (RESTful Web Service)
Management Module
Asynchronous Event Handler (RESTful Web Service)
Management Client (REST Client)
POST
GET/PUT
Asynchronous Event Sender (REST Client)
Management Agent (RESTful Web Service)
Management Backend Interface
OS or Embedded OS
Databases
Retrieval or Control Request/Response

Figure 1. CMS Architecture

CMS Architecture

- The followings are basic components of manager systems:
  - Frontend Interface
  - Asynchronous Event Handler
  - Management Client
  - Management Module

Frontend Interface

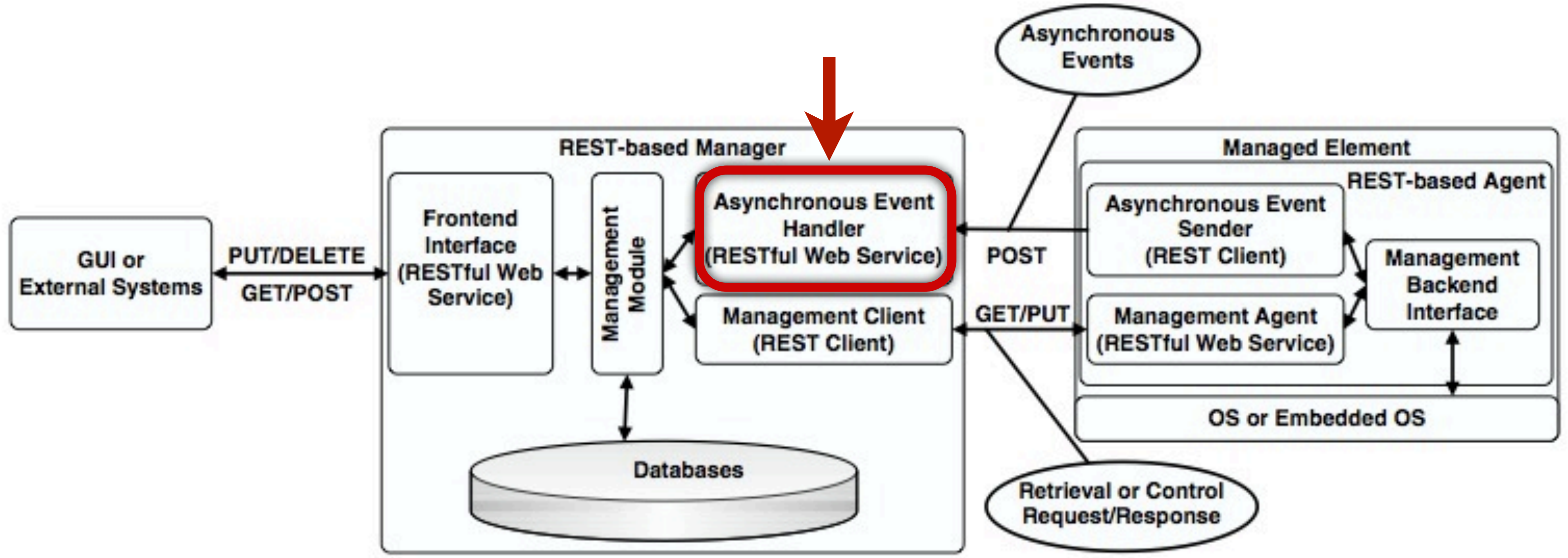


Asynchronous Events
REST-based Manager
Managed Element
REST-based Agent
GUI or External Systems
PUT/DELETE
GET/POST
Frontend Interface (RESTful Web Service)
Management Module
Asynchronous Event Handler (RESTful Web Service)
Management Client (REST Client)
POST
GET/PUT
Asynchronous Event Sender (REST Client)
Management Agent (RESTful Web Service)
Management Backend Interface
OS or Embedded OS
Databases
Retrieval or Control Request/Response

Figure 1. CMS Architecture

Frontend Interface receives requests from the user interfaces, passes them to the **Managment Module** and returns response.

Asynchronous Event Handler

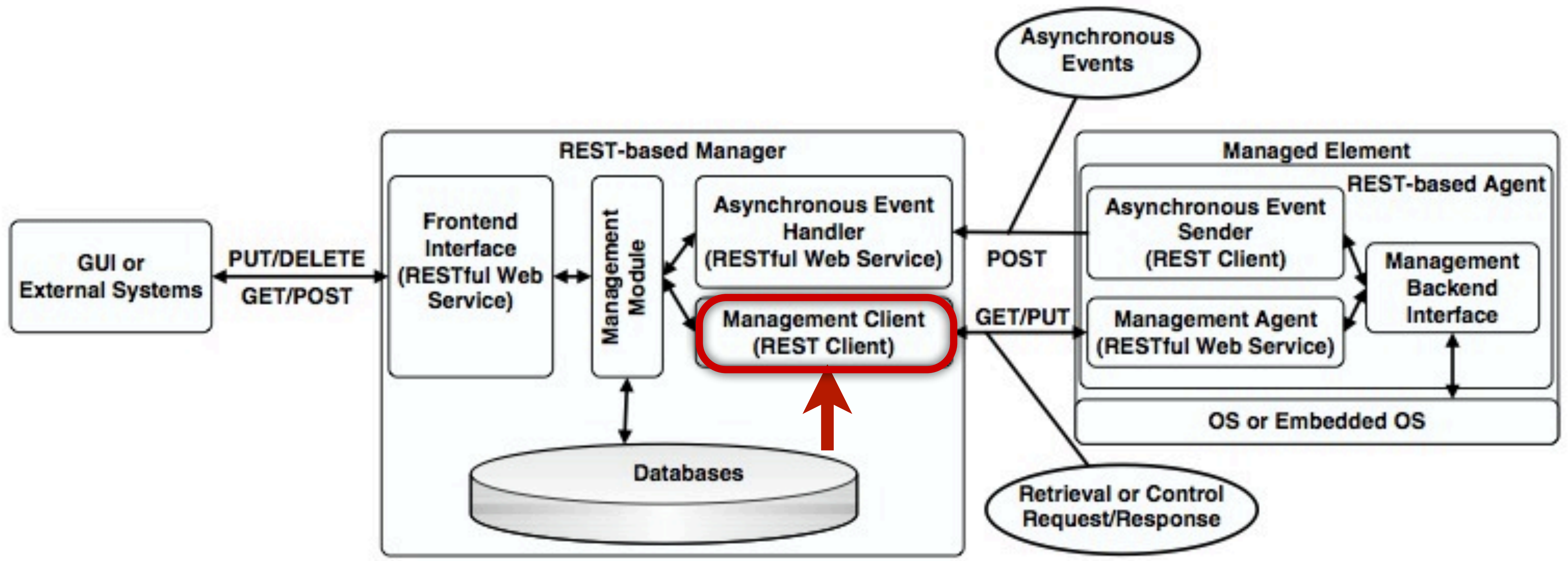


Asynchronous Events
REST-based Manager
Managed Element
REST-based Agent
GUI or External Systems
PUT/DELETE
GET/POST
Frontend Interface (RESTful Web Service)
Management Module
Asynchronous Event Handler (RESTful Web Service)
Management Client (REST Client)
POST
GET/PUT
Asynchronous Event Sender (REST Client)
Management Agent (RESTful Web Service)
Management Backend Interface
OS or Embedded OS
Databases
Retrieval or Control Request/Response

Figure 1. CMS Architecture

Asynchronous Event Handler receives notifications form managed elements, stores notification to the **Manage Module**.

Management Client

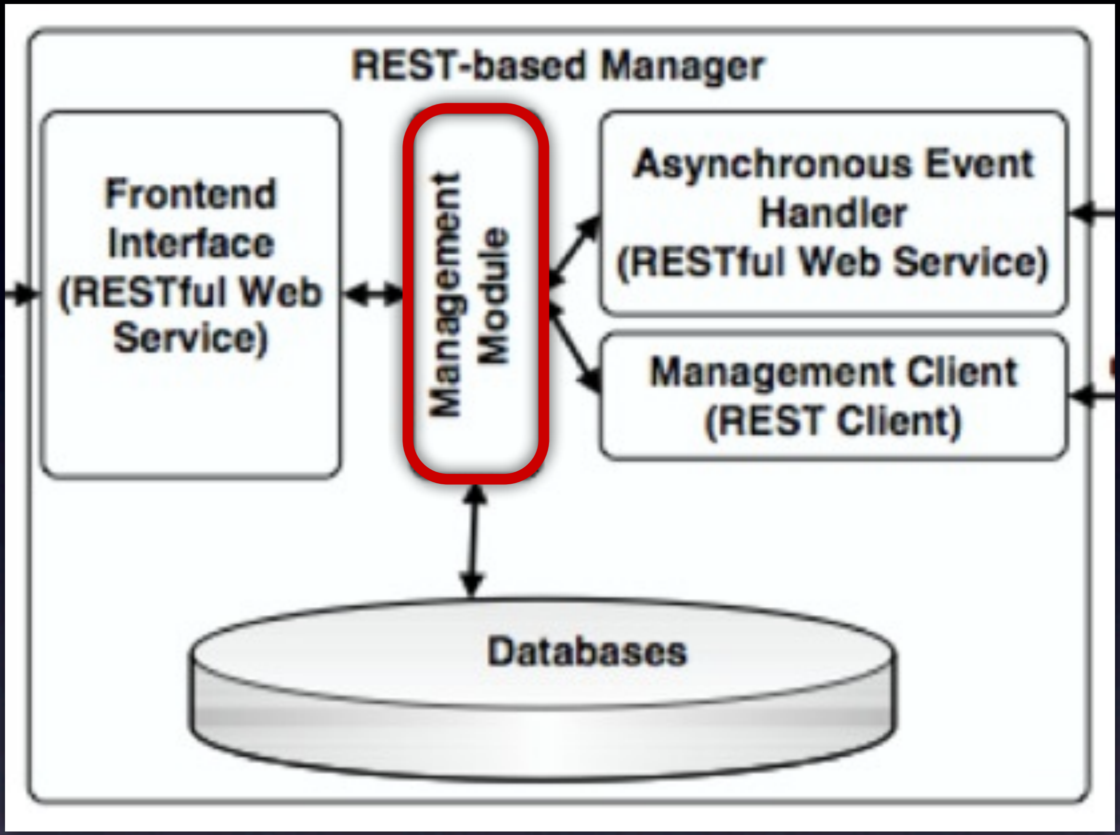


Asynchronous Events
REST-based Manager
Managed Element
REST-based Agent
GUI or External Systems
PUT/DELETE
GET/POST
Frontend Interface (RESTful Web Service)
Management Module
Asynchronous Event Handler (RESTful Web Service)
Management Client (REST Client)
POST
GET/PUT
Asynchronous Event Sender (REST Client)
Management Agent (RESTful Web Service)
Management Backend Interface
OS or Embedded OS
Databases
Retrieval or Control Request/Response

Figure 1. CMS Architecture

Management Client collects information from managed elements periodically and stores information in database. It also sends a control message to managed elements.

Management Module


REST-based Manager
Frontend Interface (RESTful Web Service)
Management Module
Asynchronous Event Handler (RESTful Web Service)
Management Client (REST Client)
Databases

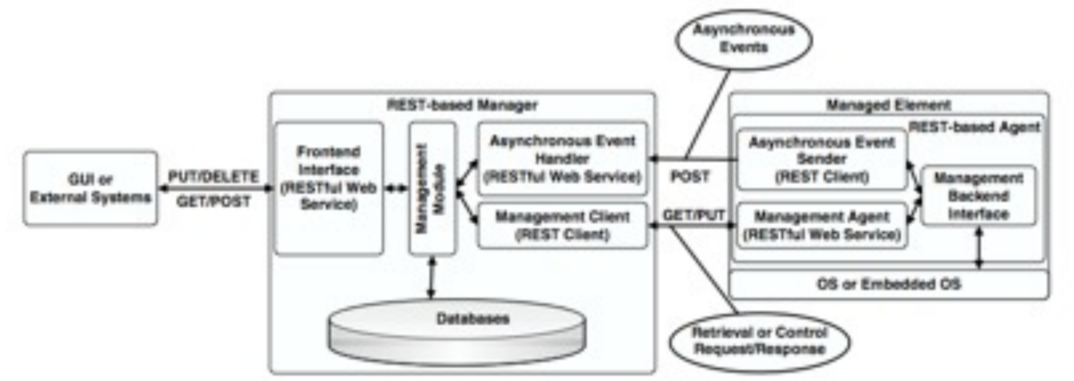
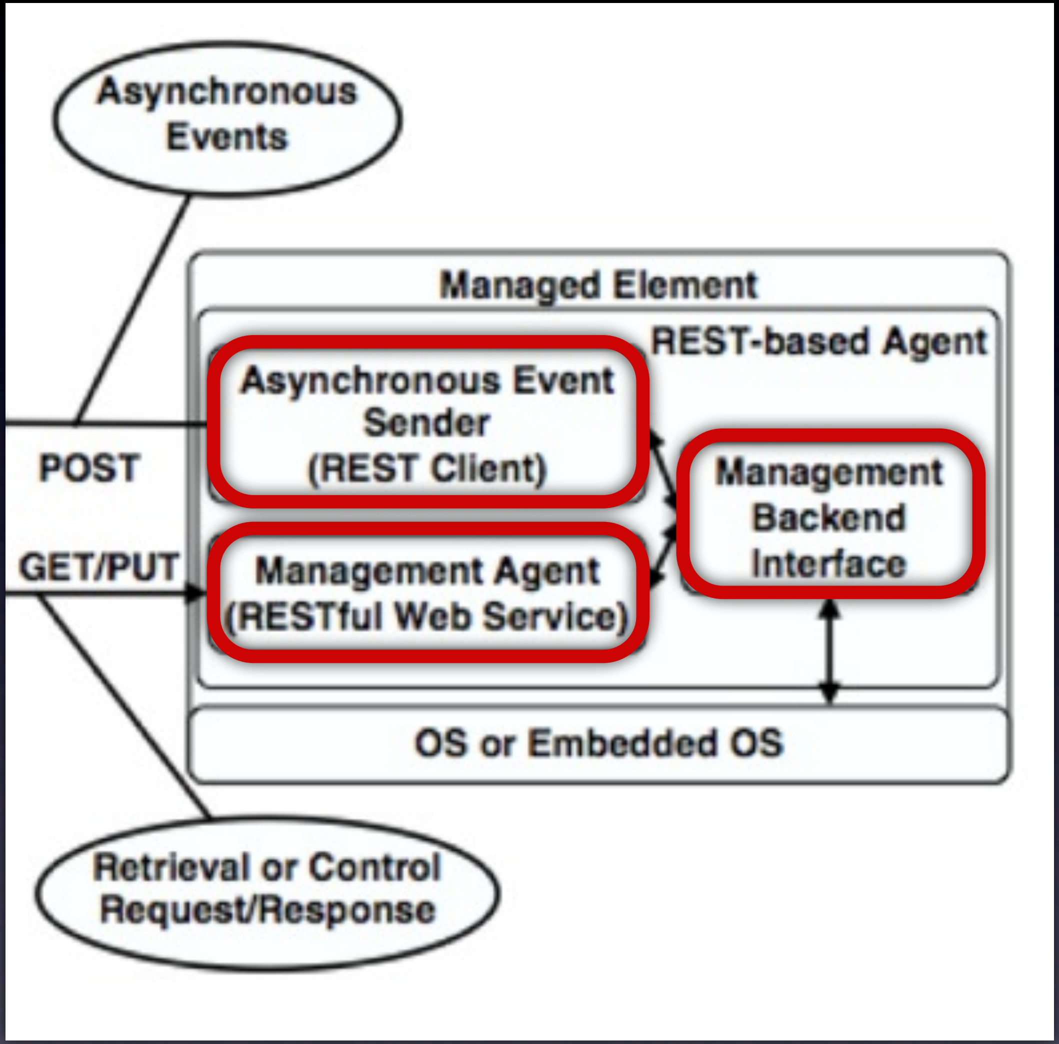

Asynchronous Events
GUI or External Systems
PUT/DELETE
GET/POST
REST-based Manager
Frontend Interface (RESTful Web Service)
Management Module
Asynchronous Event Handler (RESTful Web Service)
Management Client (REST Client)
Databases
POST
GET/PUT
Managed Element
REST-based Agent
Asynchronous Event Sender (REST Client)
Management Agent (RESTful Web Service)
Management Backend Interface
OS or Embedded OS
Retrieval or Control Request/Response

Figure 1. CMS Architecture

Management Module

- Manages the configuration parameters for managed elements, handles logical or physical topologies of managed elements.
- Obtains monitoring information from databases or the **Management Client**
- Logs the necessary data
- Manages notification data from the **Asynchronous Event Handler**

Design of the REST-based Agent

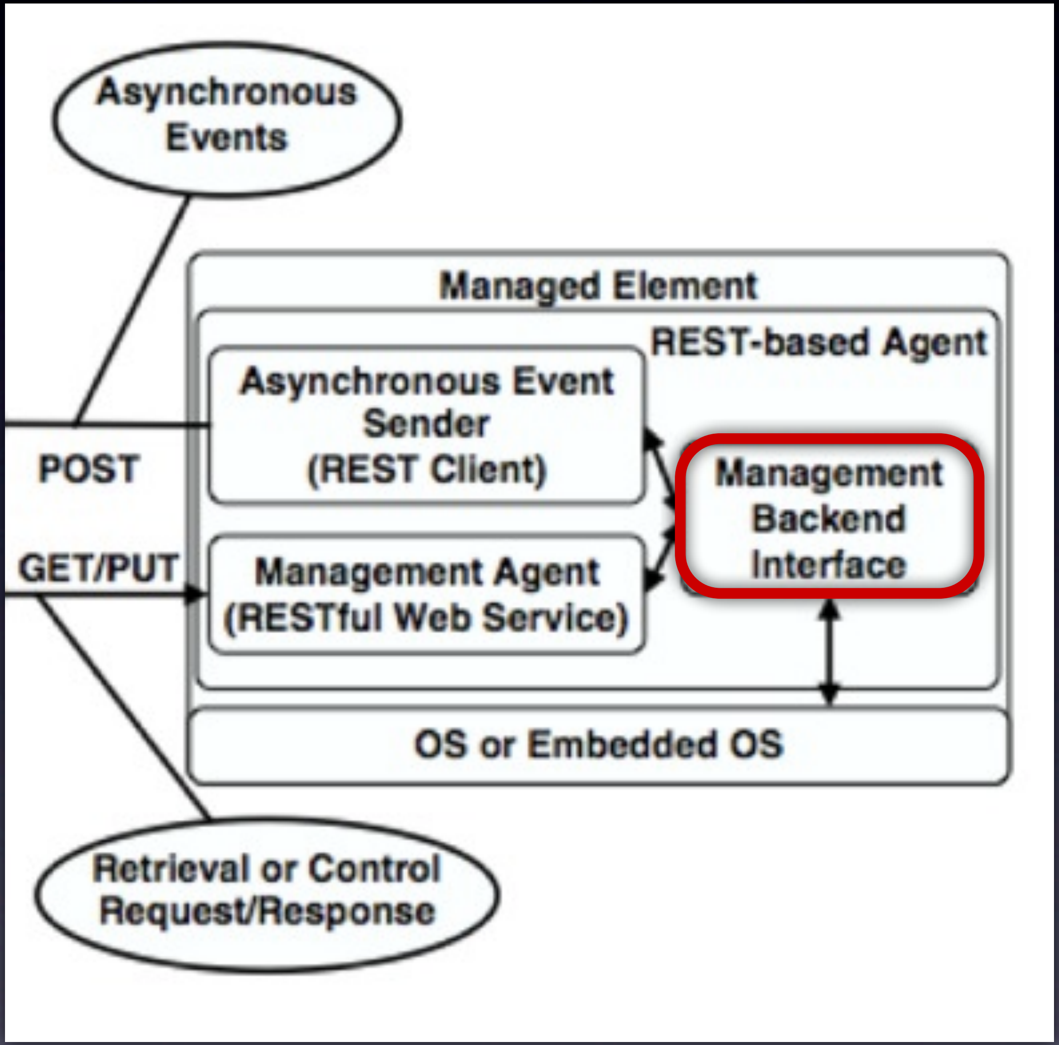

Asynchronous Events
Managed Element
REST-based Agent
Asynchronous Event Sender (REST Client)
POST
GET/PUT
Management Agent (RESTful Web Service)
Management Backend Interface
OS or Embedded OS
Retrieval or Control Request/Response

The basics components of agent systems are:

- Management Agent
- Asynchronous Event Sender
- Management Backend Interface

Management Backend Interface

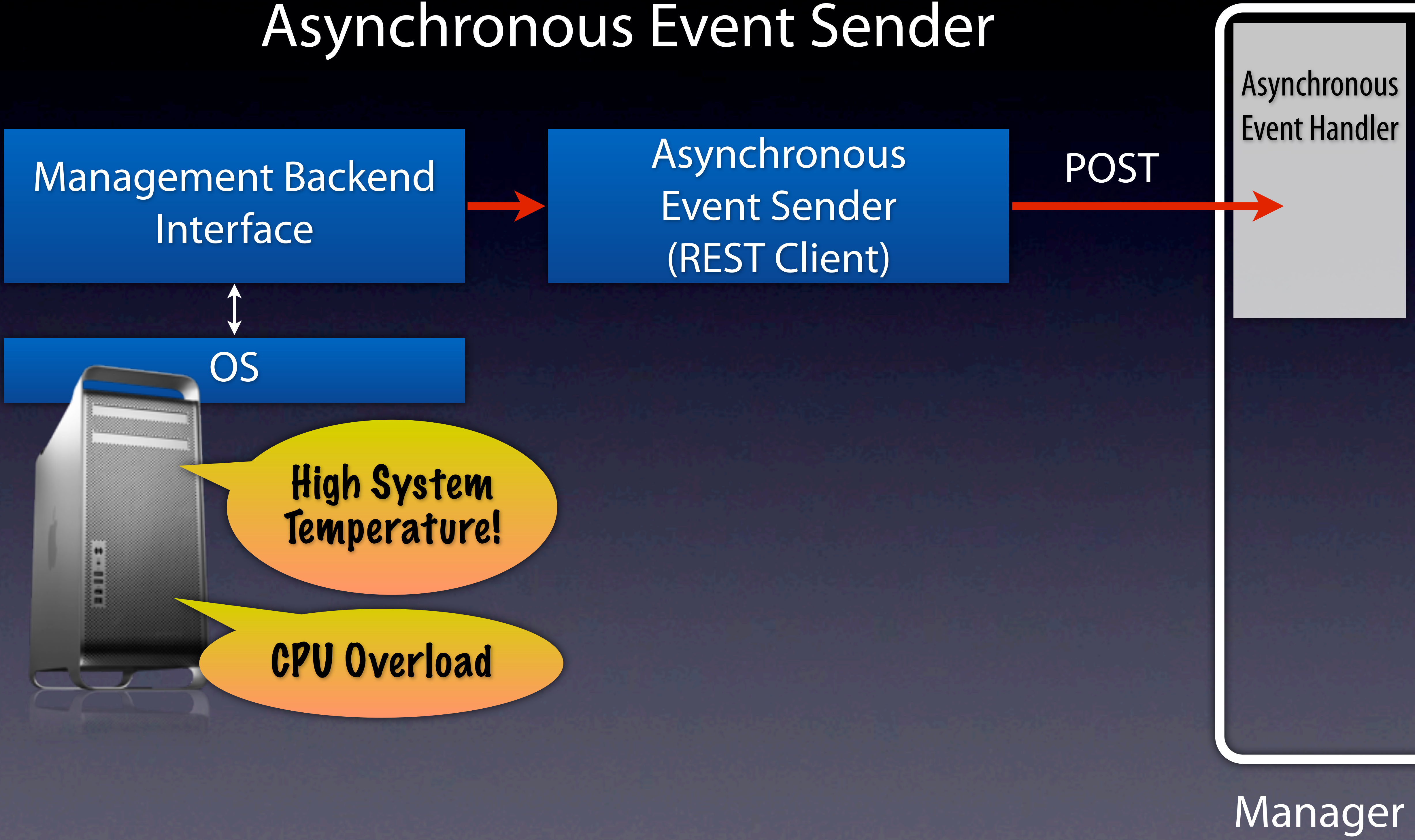
Management Backend Interface provides the **Management Agent** and **Asynchronous Event Sender** with a Management Interface.



Asynchronous Events
Managed Element
REST-based Agent
Asynchronous Event Sender (REST Client)
POST
GET/PUT
Management Agent (RESTful Web Service)
Management Backend Interface
OS or Embedded OS
Retrieval or Control Request/Response

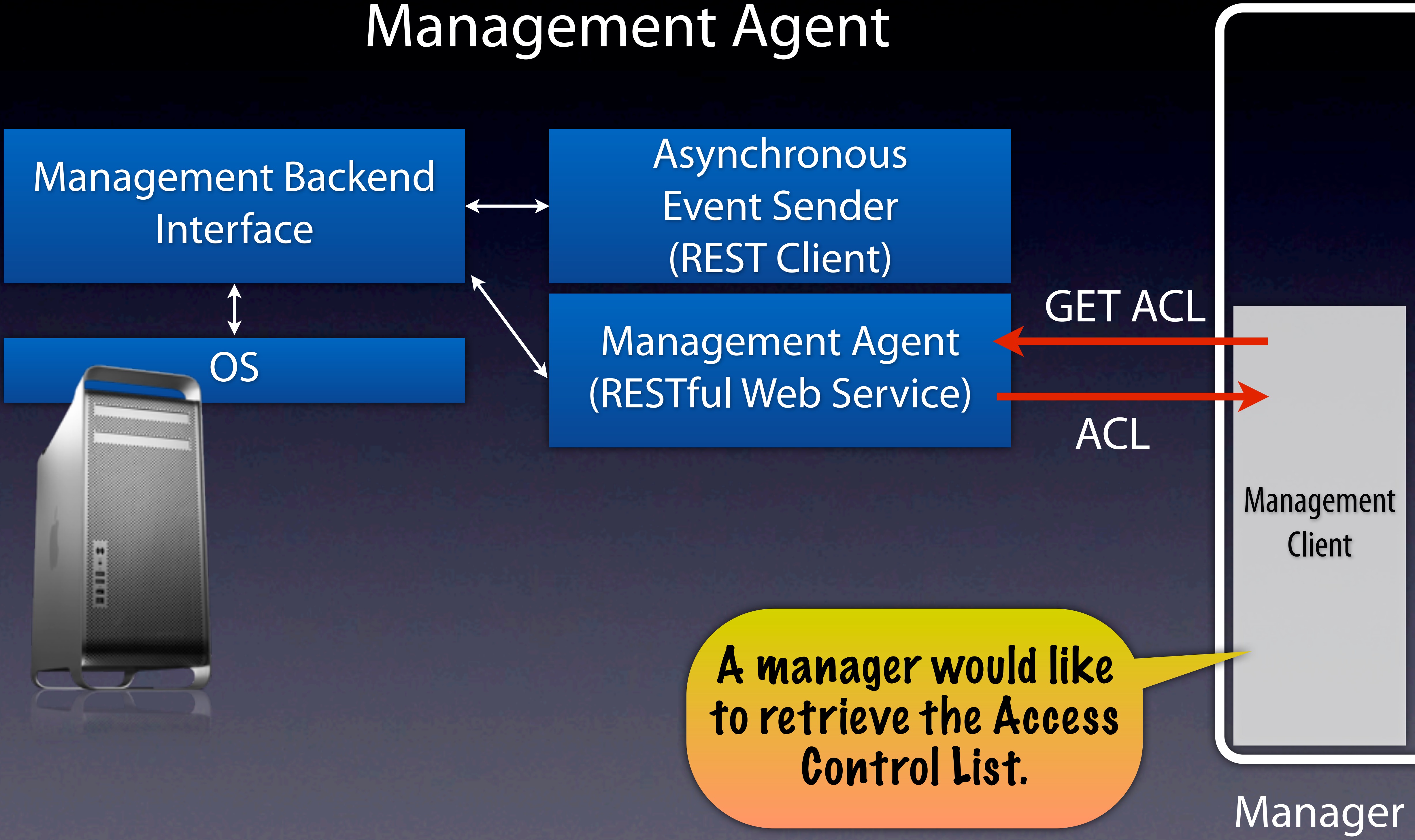
Management Backend Interface

Asynchronous Event Sender


Management Backend Interface
Asynchronous Event Sender (REST Client)
POST
Asynchronous Event Handler
OS
High System Temperature!
CPU Overload
Manager

Management Backend Interface

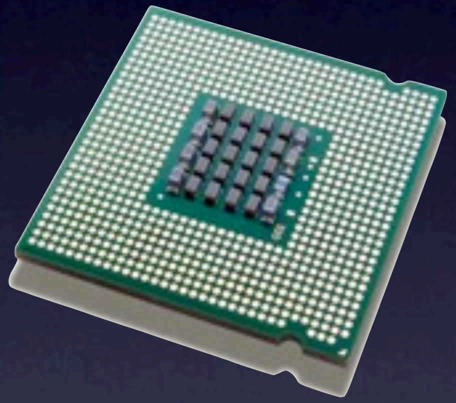
Management Agent

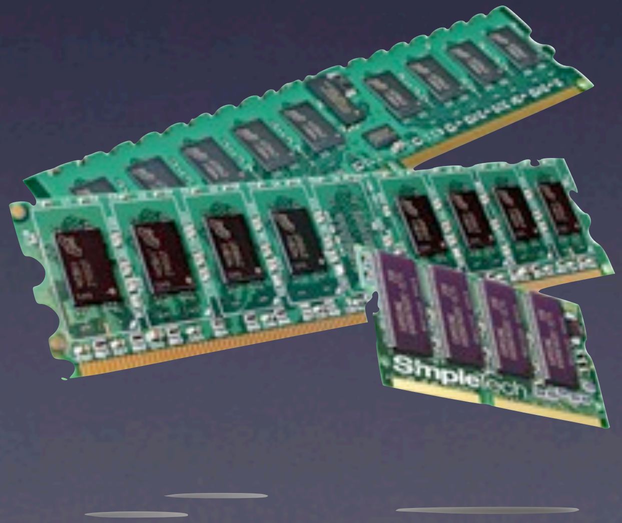


Management Backend Interface
Asynchronous Event Sender (REST Client)
OS
Management Agent (RESTful Web Service)
GET ACL
ACL
Management Client
A manager would like to retrieve the Access Control List.
Manager

Management Information Model

- Every piece of information in a RESTful Web service is represented by a URI.
- Managed element information: e.g.
  - CPU information
  - Memory information
- User-created information: e.g.
  - Group information
  - Topological information





Management Information Model

- Most management information models (MIMs) are conceptually organized as trees.
- Internal nodes of the tree represent subdivision by organization or function.
- Each variable value is stored in the corresponding leaf.
- The children of a node are numbered sequentially from left to right giving a unique name to every node.

Management Information Model

- Each node in the tree for the MIM of an agent represents management information.
- Management information is divided into three parts:

1. **Common information**
   Device configuration, Administrator list, Contract information, etc.
2. **Standard information**
   According to standard specifications
3. **Private information**

Example

The Management Client wants to retrieve the administrator list and the system description of “cloud01.snu.ac.kr”

It sends HTTP GET requests to...

```
http://cloud01.snu.ac.kr/deviceInfo/ci/AdminList
```

```
http://cloud01.snu.ac.kr/deviceInfo/smi/mib/iso/
org/dod/internet/mgmt/mib/system/sysDesc
```

Resource groups

- Many resources are required for cloud computing so they can be divided into several groups.
- Each group can be divided into several subgroups, the topology structure of the cloud infrastructure appears as hierarchy.

Resource groups

Convert the hierarchy of groups into a tree structure.

▼

Map each node of the tree to a corresponding URI.

- For example, the URI string of "WebServer#1" is "Group#1/Subgroup#2/WebServer#1"
- To get the admin list from WebServer#1:

```
http://cloud01.snu.ac.kr/Group#1/Subgroup#1/
WebServer#1/deviceInfo/ci/AdminList
```

Conclusion

- RESTful Cloud Management System (CMS) is proposed in this paper.
- The CMS fully utilizes fundamental Web technologies, such as HTTP, URIs to perform infrastructure management.
- REST-based manager and agent have been developed to achieve a pure and simple REST-based management system.

References

- H. Han, S. Kim, H. Jung et. al. A RESTful Approach to the Management of Cloud Infrastructure.
- C. Pautasso , O. Zimmermann, and F. leymann. RESTful Web Services vs. “Big” Web Services: Making the Right Architectural Decision.
- C. Pautasso. SOAP vs. REST Bringing the Web back into Web Services.

Thank You