CHEMNITZ UNIVERSITY
OF TECHNOLOGY

HP UDC – Standardizing and Automizing Data Center Operations

Ralf König

Workshop Network and Service Infrastructures,
Löbsal. April 20, 2004

Advisors: Prof. Winfried Kalfa, *TU Chemnitz*
Dr. Sven Graupner, *HP Labs, Palo Alto, CA*

Motivation

- Problems in Data Centers *(Rechenzentren)*
  - Manual operations by system administrators.
  - Limited automation with few standards
  - Overprovisioning
  - Demanding tasks:
    - Adding new hardware or software
    - Migrations
    - Reorganizing network topology

HP Utility Data Center (UDC)

- Set of Resources
  - Diskless servers with Fiberchannel controllers
  - Storage array with SAN
  - Network Appliances
    - (switch, router, load balancer, fw, ...)
- Management & Support
  - Management Rack
  - Operations Center Rack (HP OpenView)
  - Backup Rack

UDC at HP Labs, Palo Alto

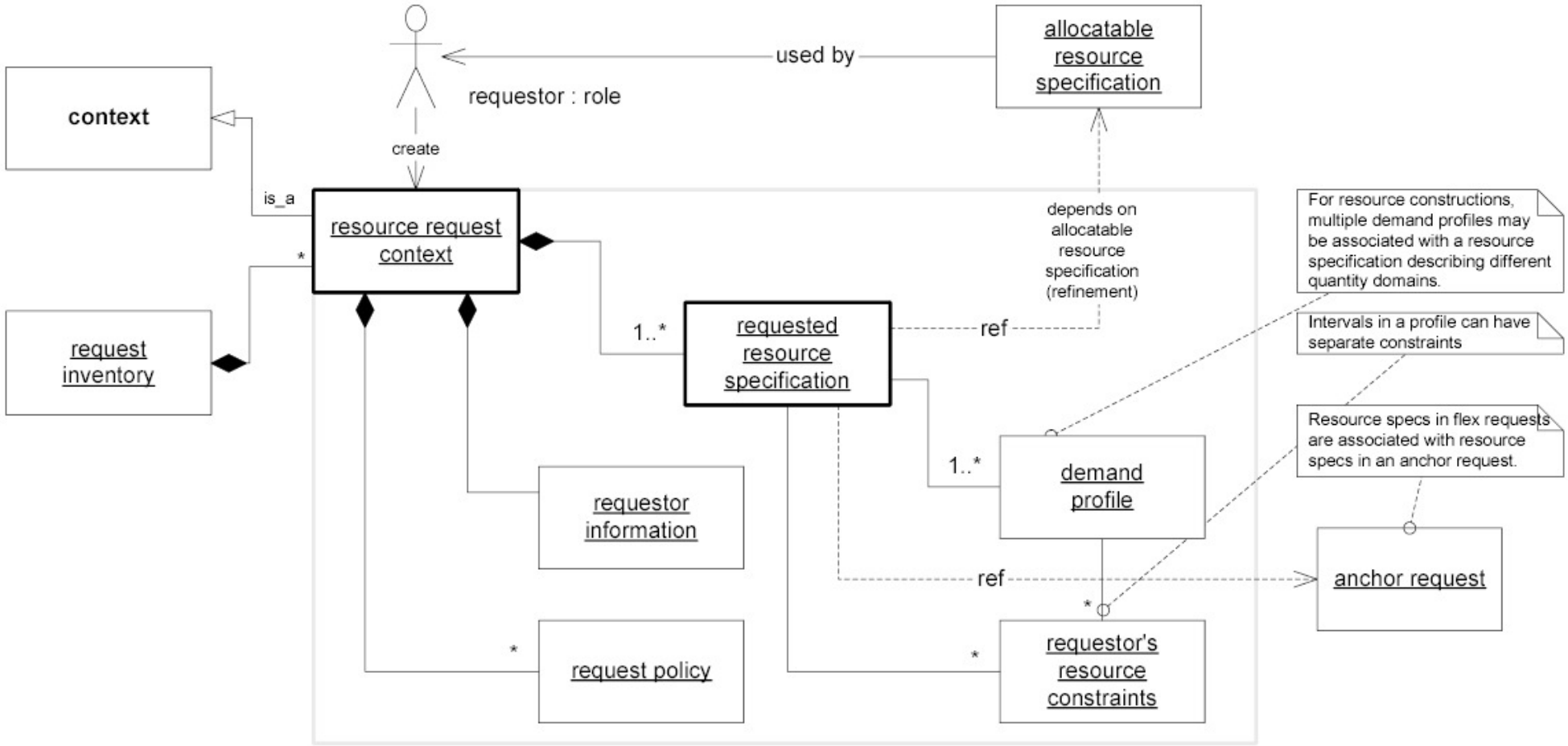
UDC: Benefits

- Flexible creation of „farms“ in a data center
- Decoupling applications from machines
- Virtualization allows standardization and consolidation
- automation and hiding of internal management operations
- Software and configuration management

UDC: Weak points of Controller Software

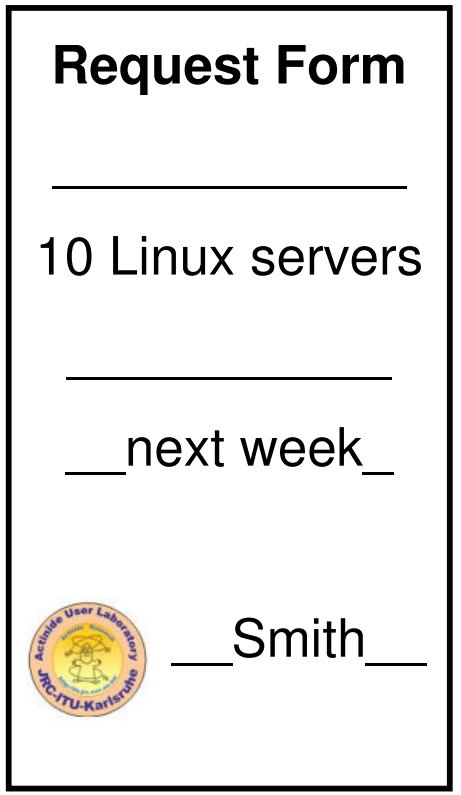
- Utility Controller user interface is completely graphical
- Limited support for virtual or transitional devices
- None of the evolving “Grid” standards have been incorporated yet
- **Fix set of resource types**
- **No planning capabilities**

Architecture: UML Diagrams

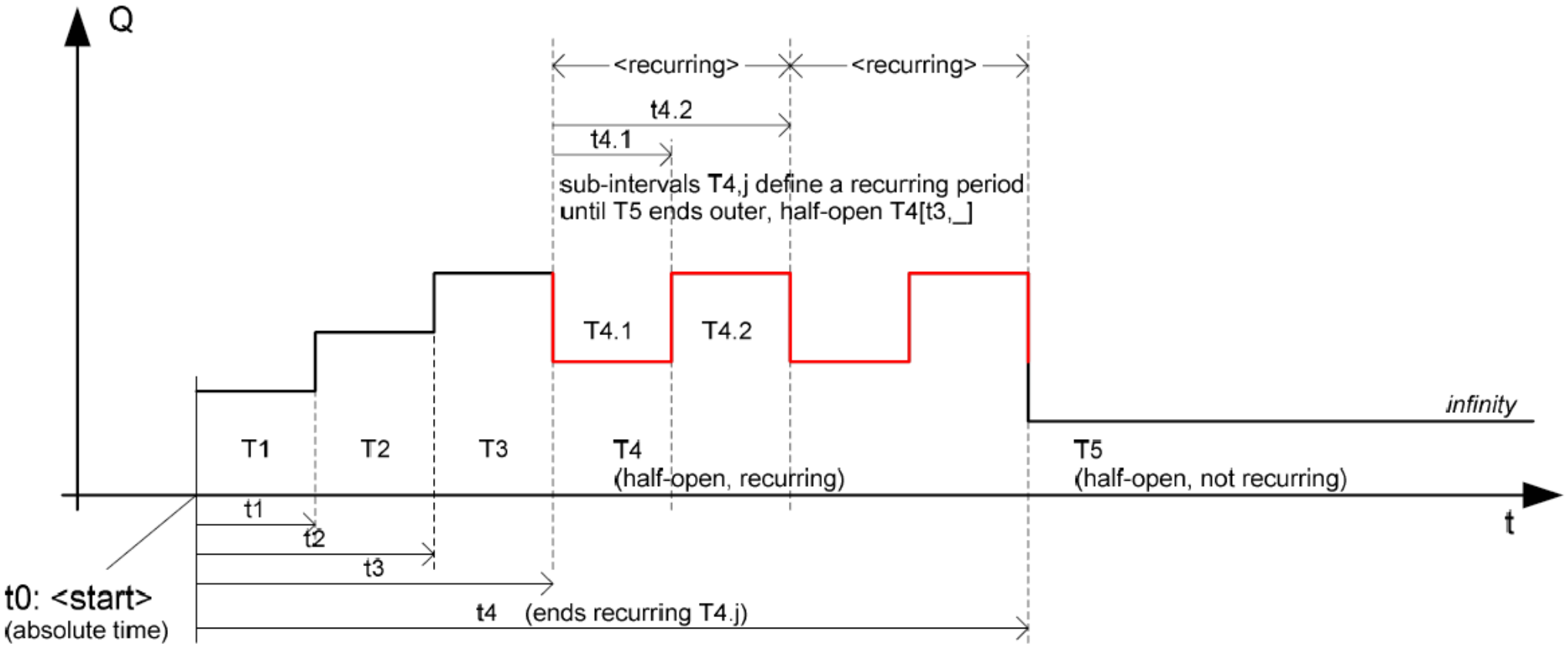


Idea: Reservation System

- Information Model describes
  - **Request** is a container for:
    - MetaData, Resource Descriptions, Profiles
  - **Resource Description** (What do I need?)
    - Properties
    - Construction
  - **Profiles** (How many? When?)
    - Quantity over Time



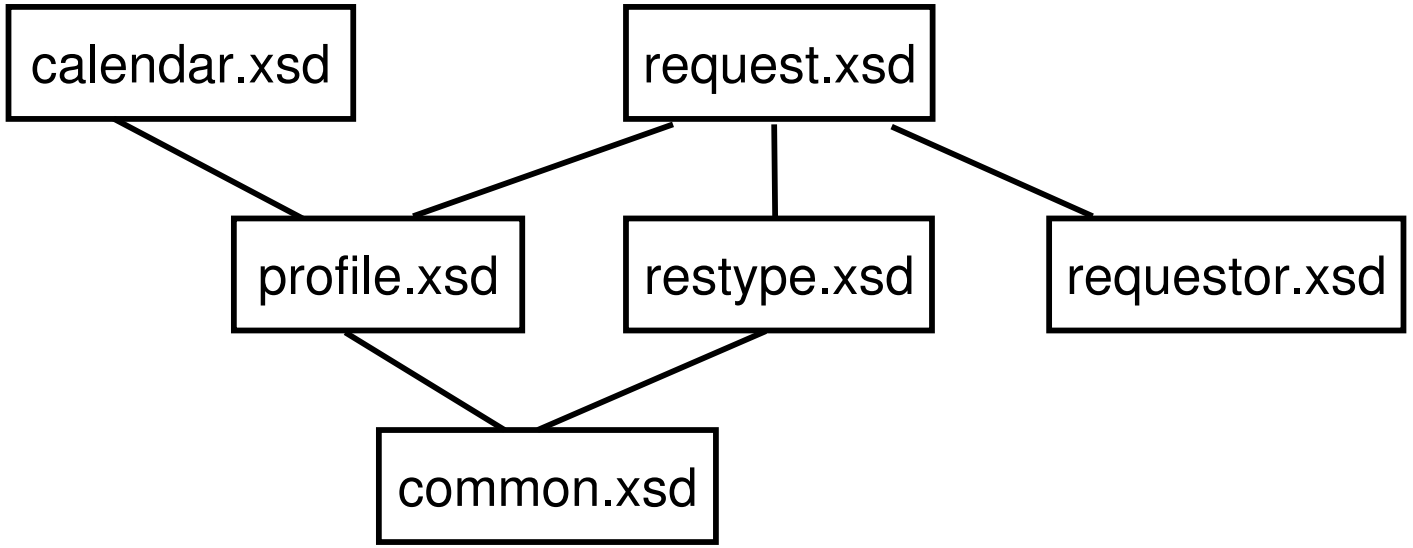
Information Model: Profile



Information Model: Resource Description

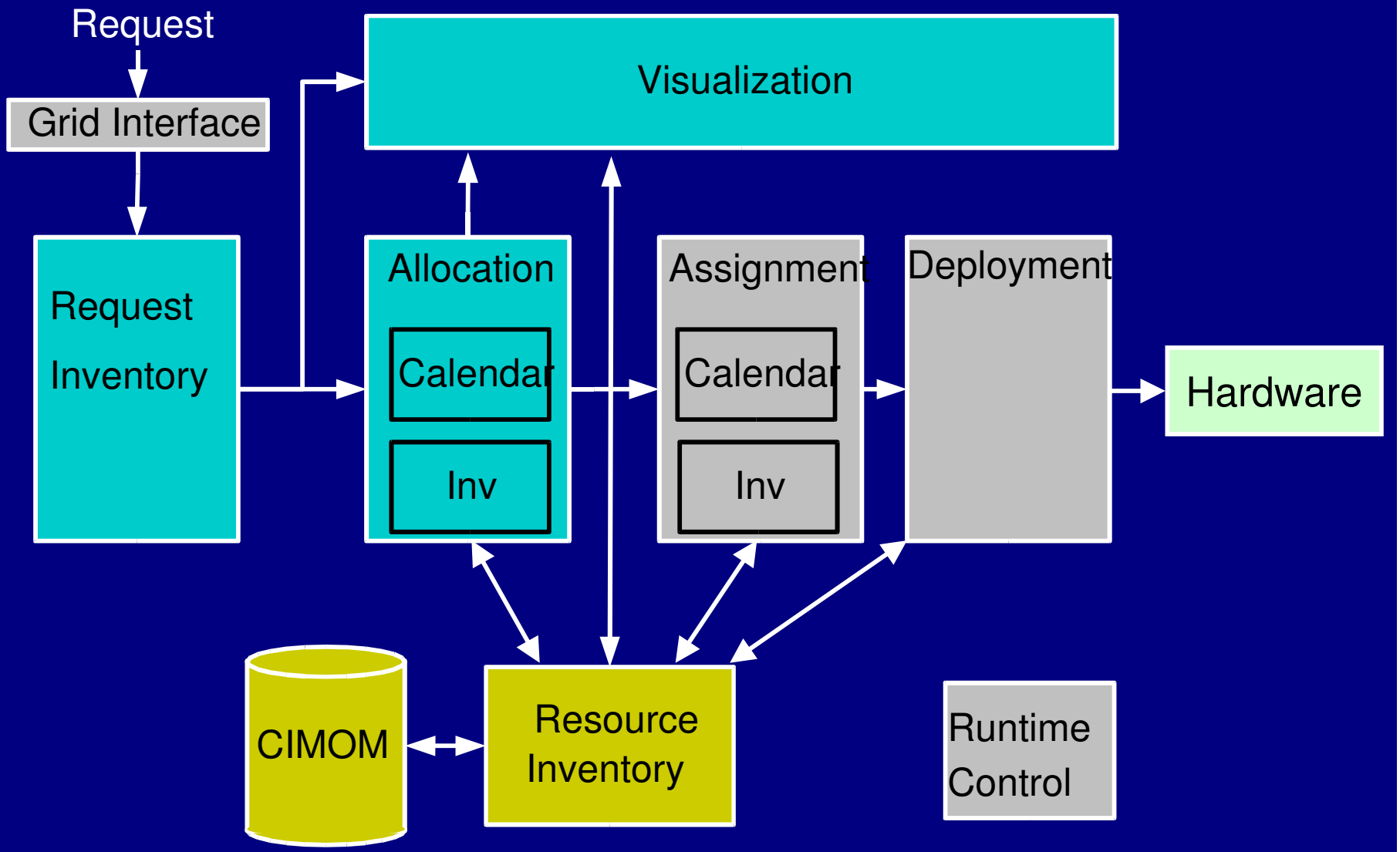
- Metadata
  - ID, name, version, creator, ...
- Associations, such as super class or construction hierarchy
- Properties
  - Attribute-Value pairs
- Constraints
  - Container for rules in policy language format

Implementation: XML Schemas

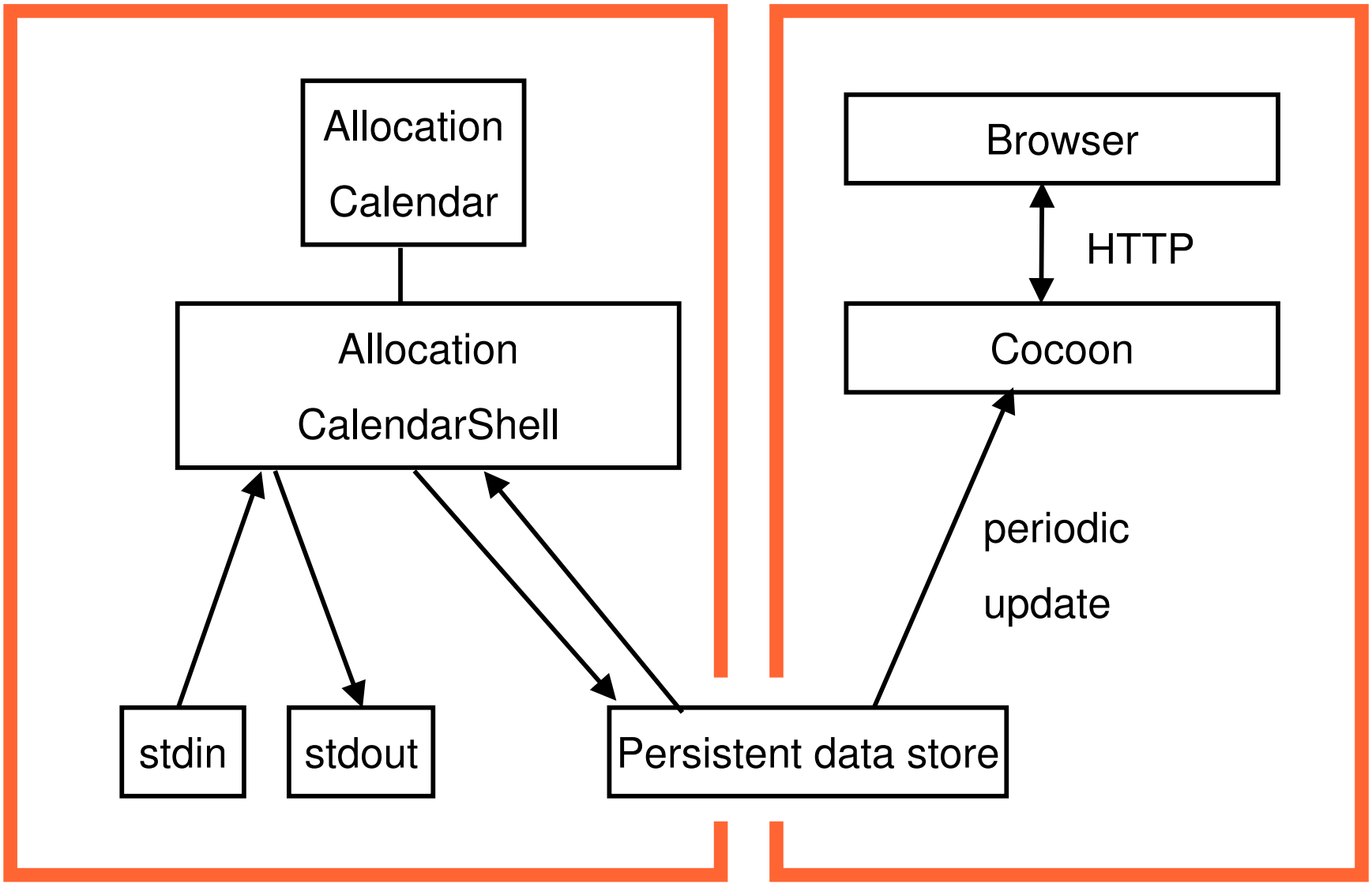


- XML schemas
- Include mechanism
- XML documents are linked

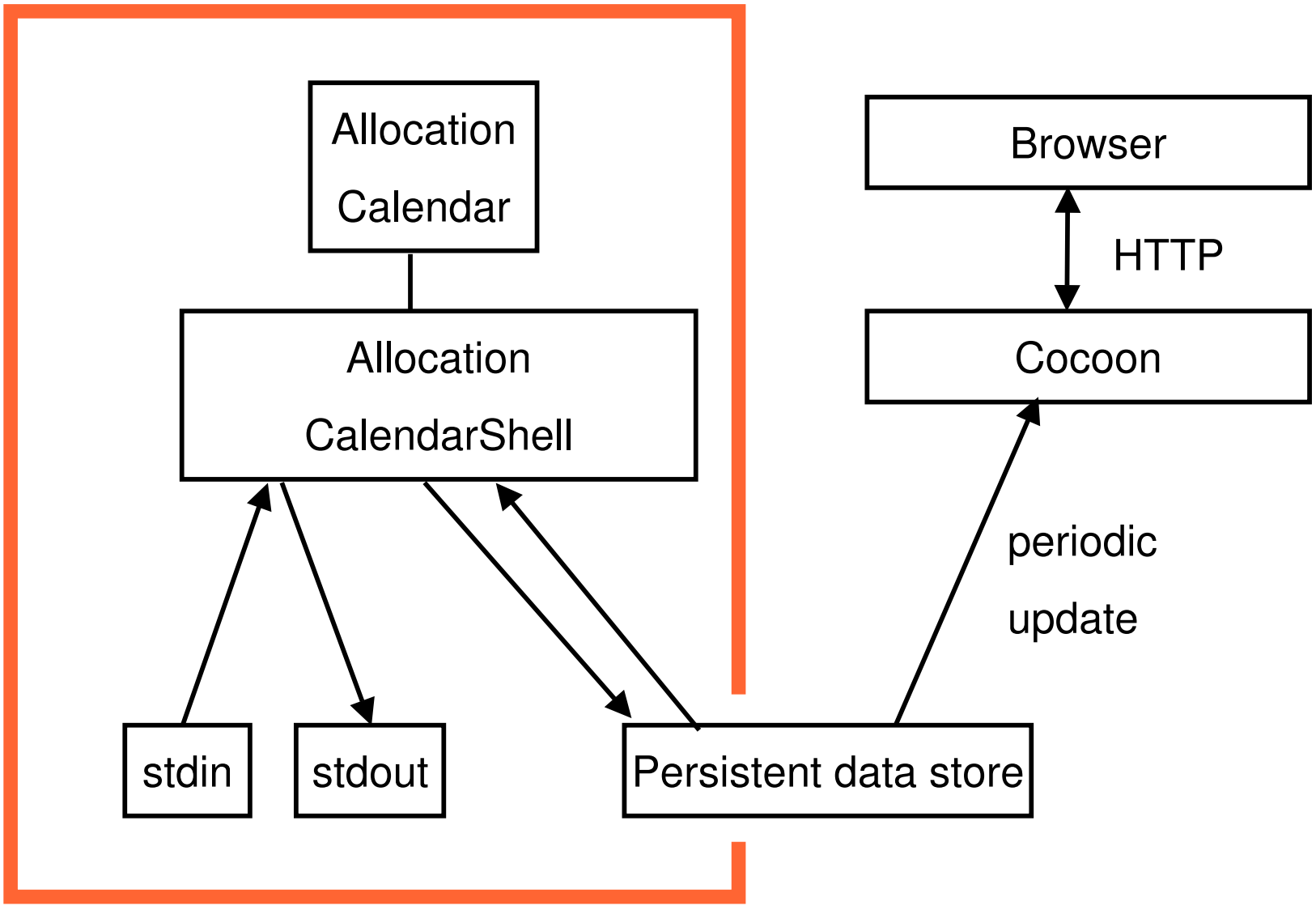
„Quartermaster“ prototype architecture



The two Interfaces: Shell and GUI



Shell Interface

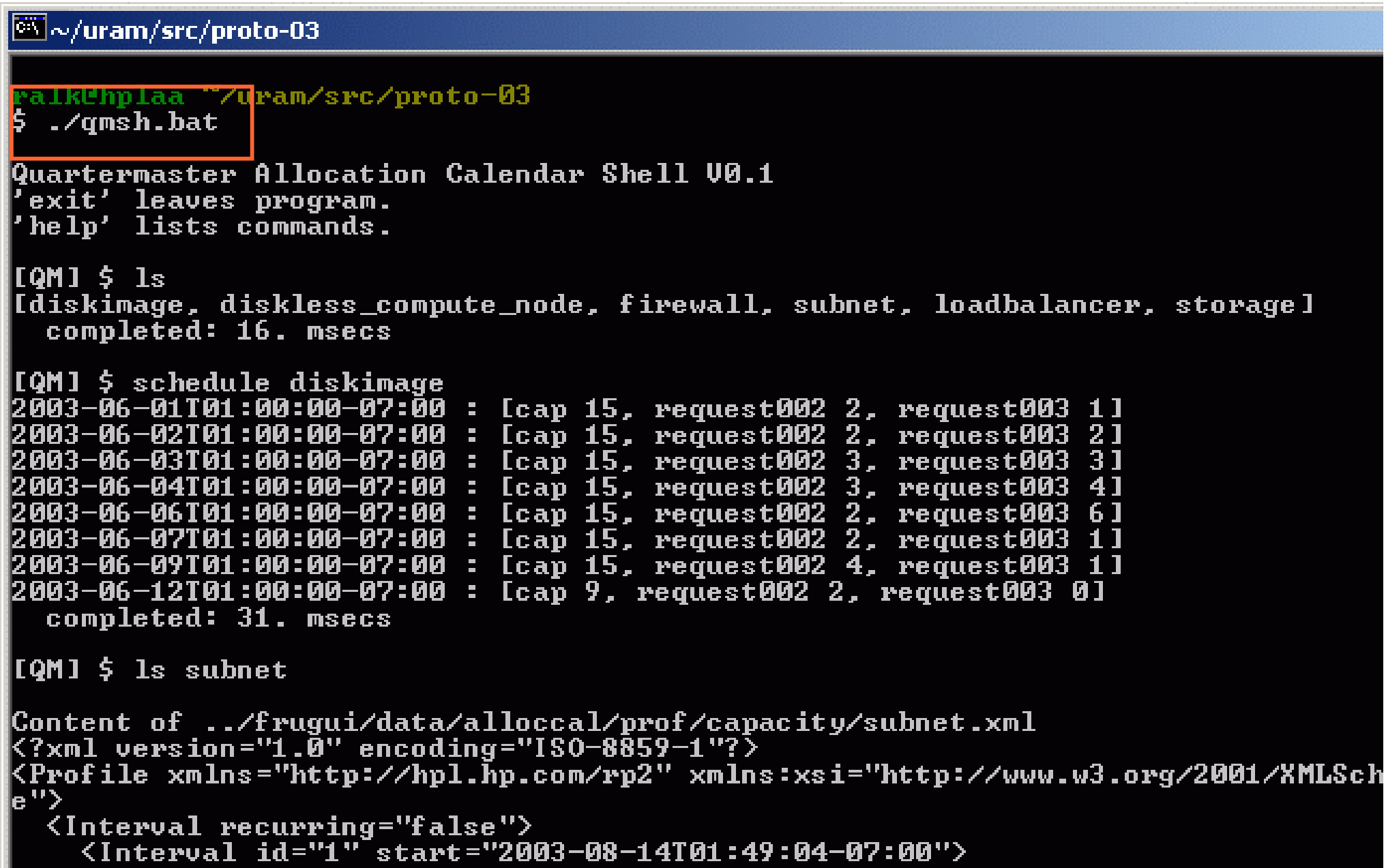


AllocationCalendarShell in Java

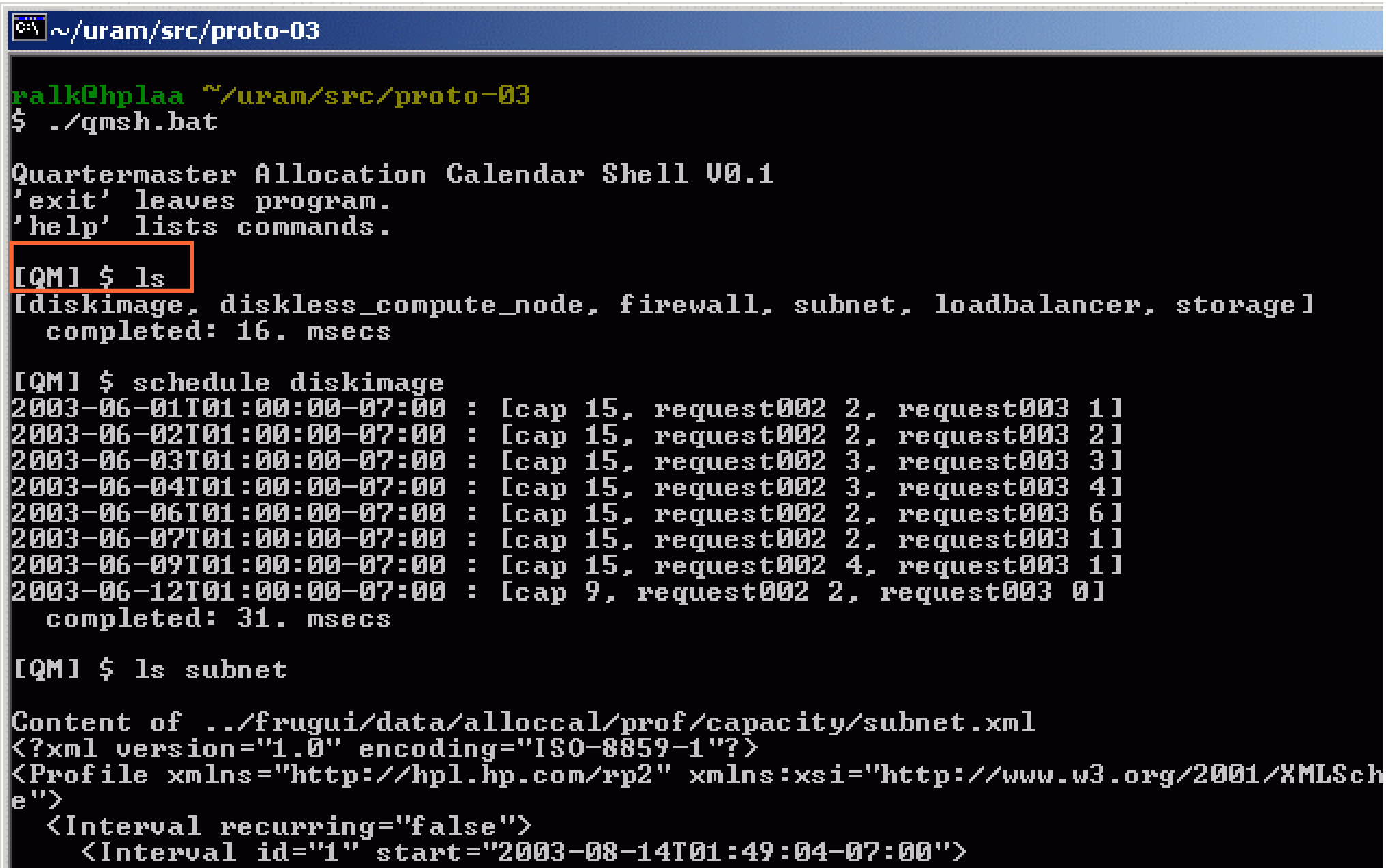
- Provides scripting capabilities for the „site calendar“

- Command set:
  - General commands
    - **`help, exit, #`**
  - Operations on the site calendar
    - **`load, save, ls`**
  - Operations on resource calendars
    - **`put, ls, schedule, save, rm`**
  - Operations on demand profiles
    - **`put, rm`**

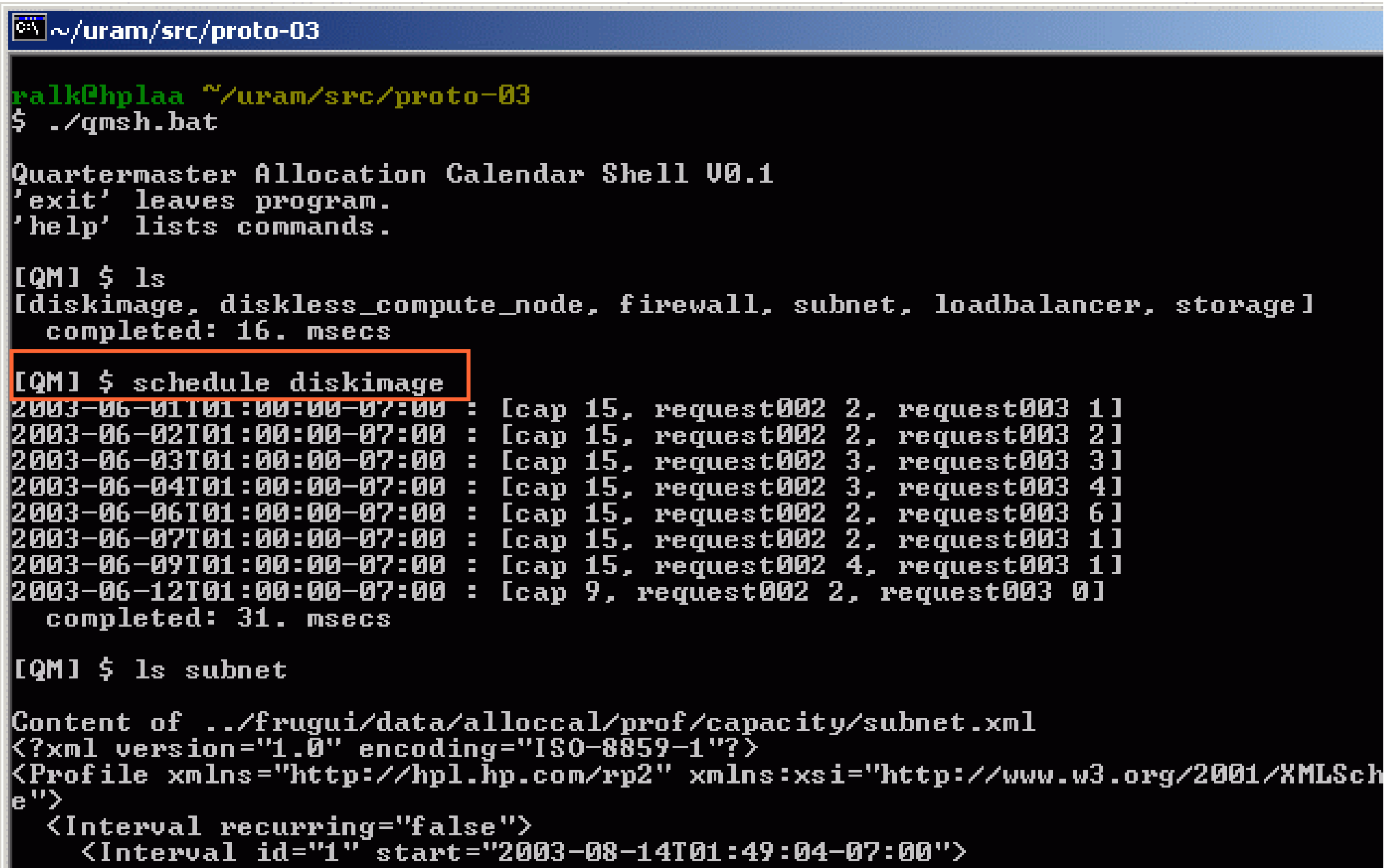
Shell Commands by Example



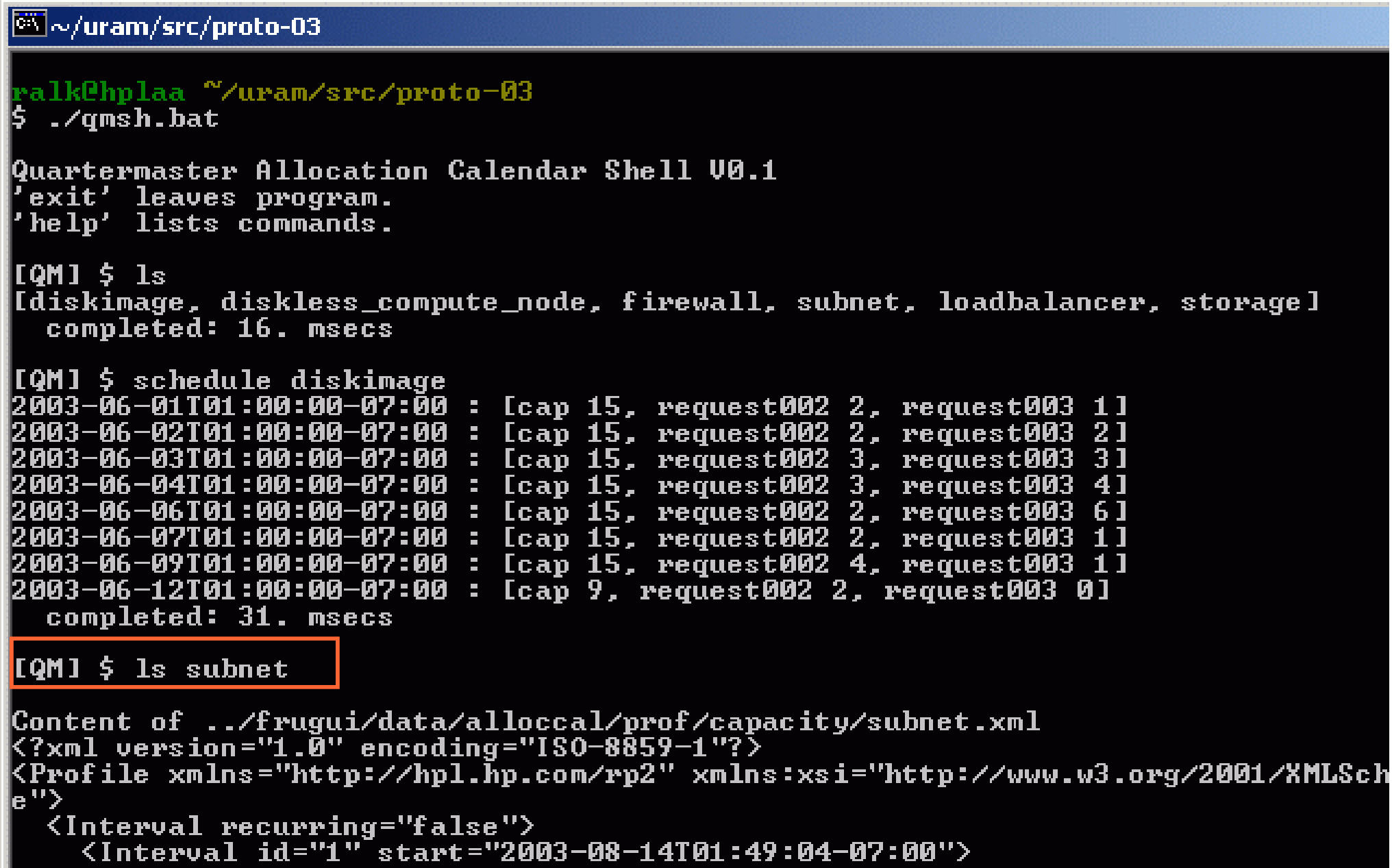
Shell Commands by Example



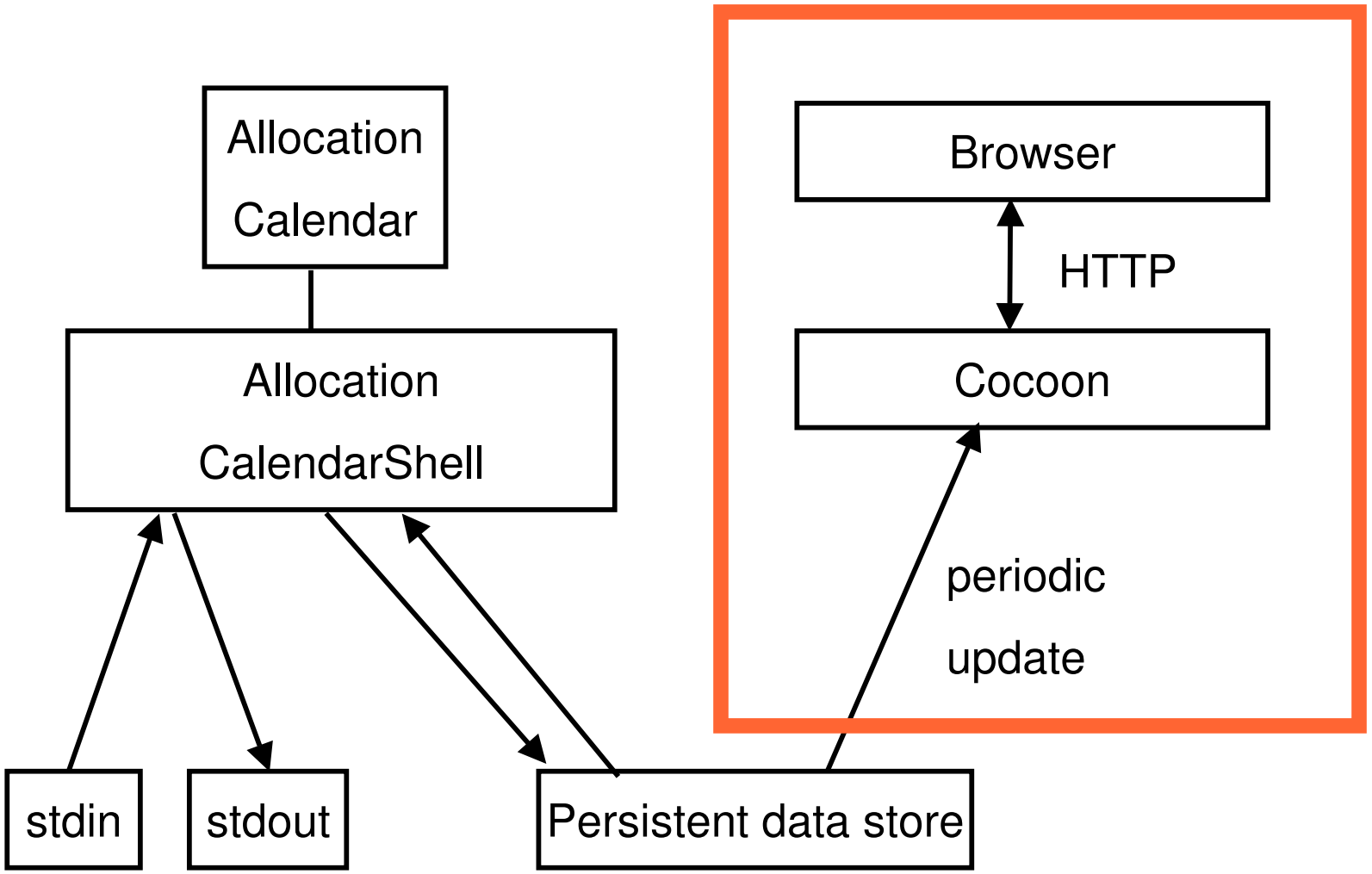
Shell Commands by Example



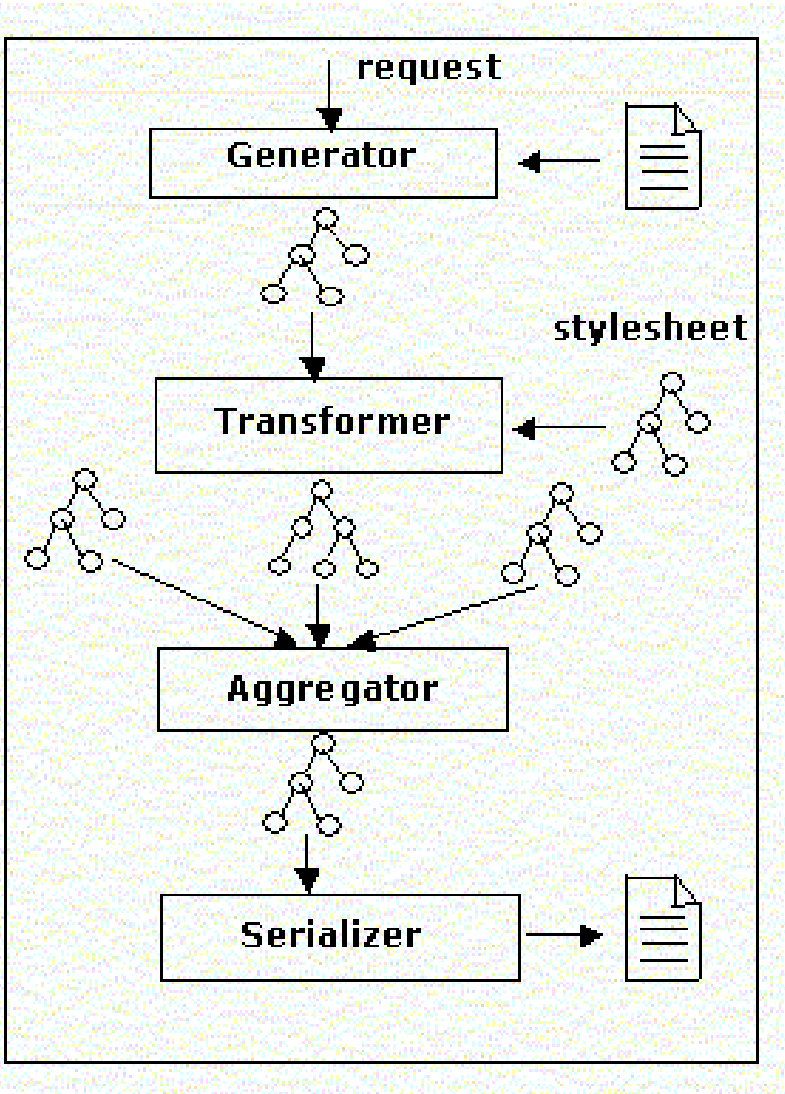
Shell Commands by Example



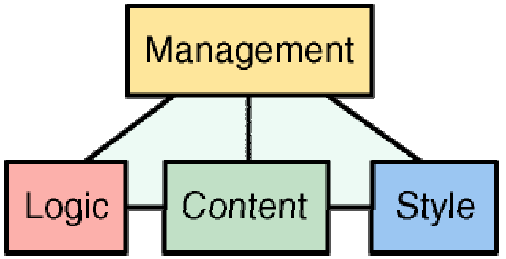
Graphical User Interface



Cocoon: A web development framework



- Generator:
  - Files (XML)
  - Query result from XML or SQL database
  - eXtensible Server Pages (XSP)
  - LDAP
- Transformer
  - XSL stylesheets
- Serializer
  - HTML, SVG, SVGtoBitmap



Cocoon: Request processing

Matchers on the HTTP Request:

- URL and its parts (host, domain, path, parameters)
- Session ID
- Cookies
- Browser type
- Referer
- Sender IP address

Combinations of the above

+ wildcards + regular expressions!

Sitemap.xmap

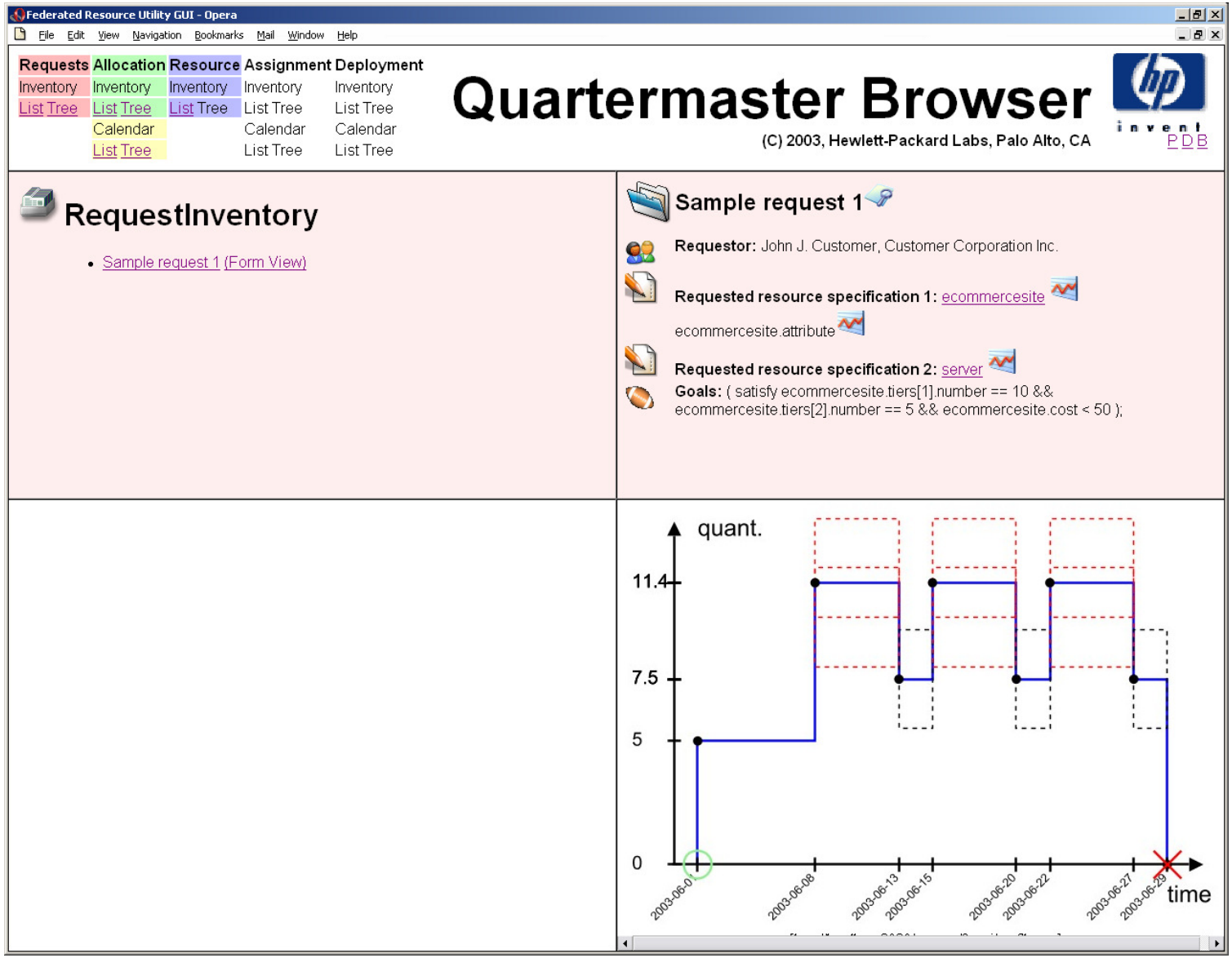
```
<map:match pattern="view">
   <map:match type="request-parameter-wildcard“ pattern="/*/**">
      <map:parameter name="parameter-name" value="data" />

         <map:match type="request-parameter-wildcard“ pattern="**">
            <map:parameter name="parameter-name" value="style" />
            <!--   {1}  = style -->
            <!-- {../1} = data -->
            <map:generate src="data{../0}.xml" />

           <map:transform src="stylesheets/{1}.xsl">
              <map:parameter name="base-url" value="/cocoon/frugui" />
              <map:parameter name="context" value="{../1}" />
           </map:transform>

           <map:serialize />
         </map:match>
   </map:match>
</map:match>
```

GUI Screenshot



Decisions and Results

- Separation of Concerns
- Data Organization: Files vs. Database
- Lack of well-established query language for XML
- SQL and XML – mapping data back and forth
- Cocoon runtime performance
- Cocoon documentation

The broader picture

Other people at HP Labs working on:

- policy integration
- agreement protocols in grid environments
- resource type specifications related to Common Information Model (CIM)
- server consolidation, probabilistic capacity planning, Quality of Service management
- workload simulation and placement optimization
- relations to other major HP products: HP Utility Data Center and HP OpenView

End

Thanks for your attention!