# APACHE KNOX

## A3 Architecture Overview

## Apache Knox A3AO Objectives

- One page architecture (external/internal views)
- Key functional requirements
- Key non-functional requirements
- Patterns/design aspects
- Technology list

## Key Functional Requirements

- REST API Gateway for Hadoop Clusters
- Single AP for all REST interactions in cluster

- Authentication (LDAP / AD Provider)
- Federated SSO (HTTP header based)
- Authorization (Service Level)
- Auditing

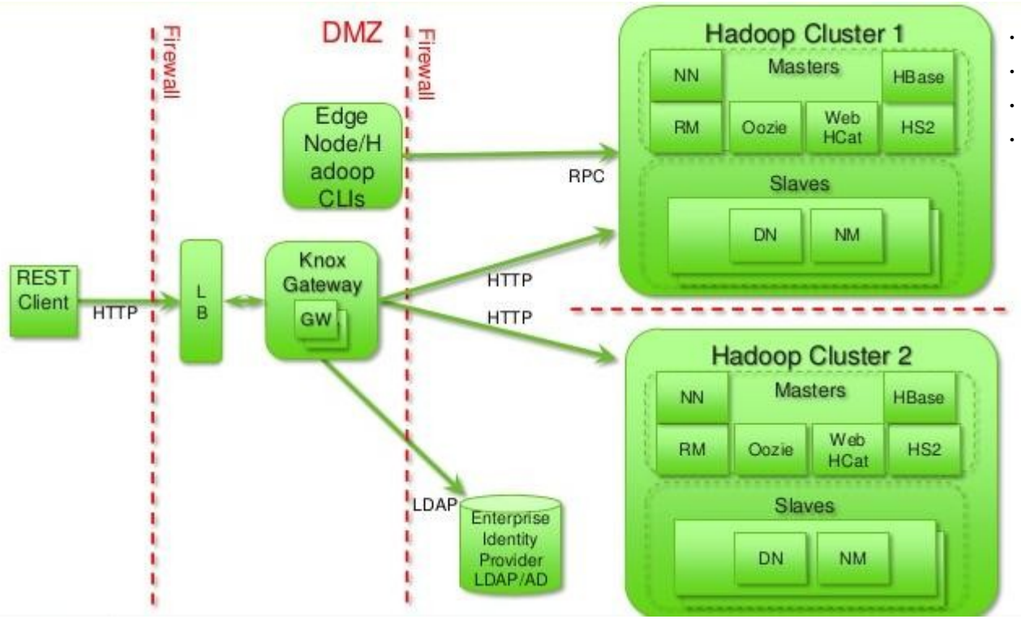
## Key Non-Functional Requirements

- Extensible (through pluggability)
- Configurable
- Performance (fast responses) on request processing

## Patterns / Design Aspects

- Reverse Proxy Gateway
- Servlet filter chain
- Perimeter Level Security
- Extension mechanisms
  - Service (new REST End Points)
  - Provider (new features for Services to use)
- Streaming during processing
- Deployment / Runtime phases

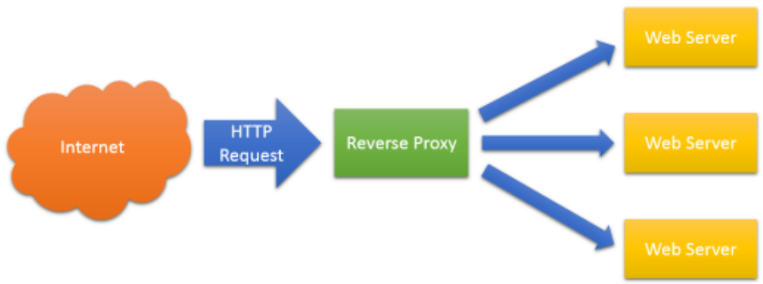
## Knox Architecture From The Outside




- Clients make requests to Knox
- Knox authenticates against EidP
- Knox Processes (more later)
- Knox sends request onto Hadoop services

## Patterns cont.

- Reverse Proxy Pattern
  - Knox is the Reverse Proxy Gateway
- Provides access to multiple, known, services
- Any client can access Hadoop services via Knox
- Side benefits
  - Single server/port for all services
  - Single point for authentication
  - Single point to secure against external threats
  - Single SSL certificate




## Technology List

- Shiro authentication provider (LDAP/AD/BASIC)
- Pac4J (authentication / authorization mechanisms)
- ACL based authorization
- Log4j (auditing)
- Embedded Jetty JEE server
- Maven multi-module build
- Topology descriptors
- Kerberos

## Version Information

| | |
|---|---|
| **Title** | Apache Knox A3AO |
| **Author** | John McParland (john.mcparland AT cgi.com / johnmmcparland AT gmail.com) |
| **Version** | 0.1 |
| **Date** | M 24 Oct 2016 |
| **Audience** | Apache Knox Developers |

## Deployment

- Convert topology to WARs
- Based on Contributors
  - Pluggable components
- Basic WAR created
- Visitor pattern
  - Contributors modify WAR
  - Adding Services/Providers to WAR
- Service Deployment
  - Svcs created for roles in descriptor
  - Deployment framework adds filters to it
- Provider Deployment
  - Prvdrs created for roles in descriptor
  - Will add runtime deployment descriptors
  - Adds Servlet Filters to gateway runtime

## Runtime

- Filter chains managed by GatewayServlet
  - Allows more powerful URL matching
- Main filter – GatewayFilter
  - Allows dynamic deployment of modified topologies

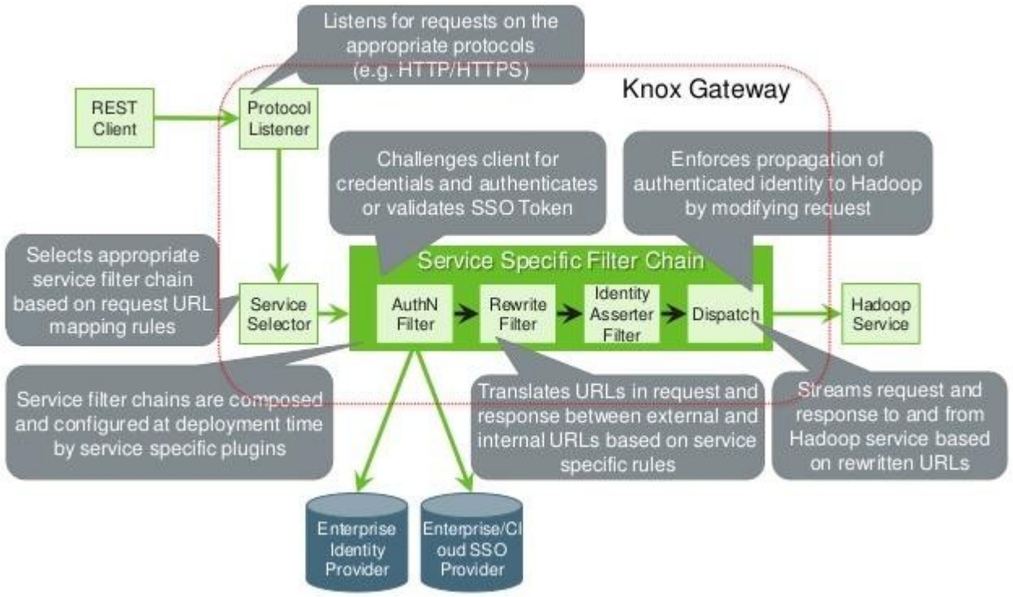
## Services

- Convert topology file to runtime descriptors
- Modify gateway or other runtime descriptors
- ServiceDeploymentContributor is interface
- Loaded via ServiceLoader mechanism
  - String lookup from topology file
- Add via service.xml and rewrite.xml
- services/service name/version/<files>
- In gateway-service-definitions module

## Providers

- Implement ProviderDeploymentContributor

## Knox Architecture From The Inside




- Come in through protocol handler
- Select appropriate service
- Service Specific Filter Chain
  - Authentication
  - Rewriting
  - Identity
  - Dispatch
- Stream in/out of Hadoop service

## Extending

- Extensions discovered via ServiceLoader (Java)
- Classpath (no recompile)
- Maven modules (add to gateway-release)
- Services (see other panel)

## Service Contribution Behaviour

- gateway.xml
  - Controls behaviour of GatewayFilter
  - Mapping of URL patterns and filter chains
- rewrite.xml
  - Rules to control URL re-writing
  - Svc Contrbtr need to provide this