

oX

oBIX Server Open Source Project

Challenges and Joys of Implementing a Lightweight
Open-Source oBIX Server

Peter Michalek

peter@michalek.org

Agenda

- Intro into oBIX
- Overview of oBIX Features
- oX Architecture
- oX Implementation
- Plugin Modules
- Security
- Performance
- Next Steps, Future
- Links/More Information
- Q/A

Intro

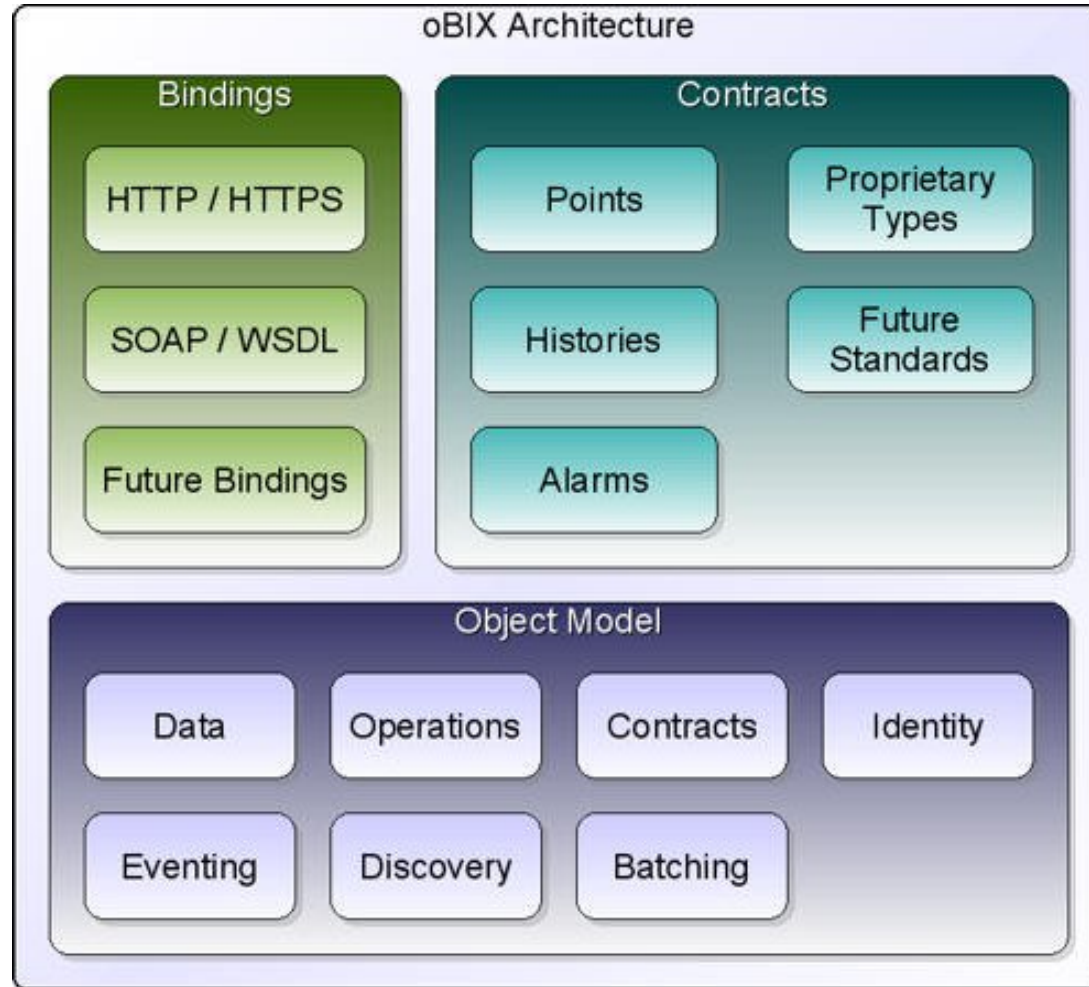
- What is oBIX
 - Modern generic device management and monitoring concepts applied to buildings space
 - The best effort in the industry to come up with:
 - simple and extensible object architecture and apply it to XML schema and http transport
 - No negative features that often plague large-committee standards
- Why use it
 - Modern Design
 - Solves challenges of extensible XML approach
 - Excellent core implementation by Brian Frank
 - Simplicity
 - Readily available open source implementation

Overview of oBIX Features

- Addresses device management
- Simple yet powerful
- REST based: easy to integrate with mashups and SOA
- Built-in alarm concept
- Built-in watch and feed
- Meta-info: version, server, etc.
- Many other things (batch, query, taxonomy)

oBIX Architecture

- From Aaron Hansen Article:

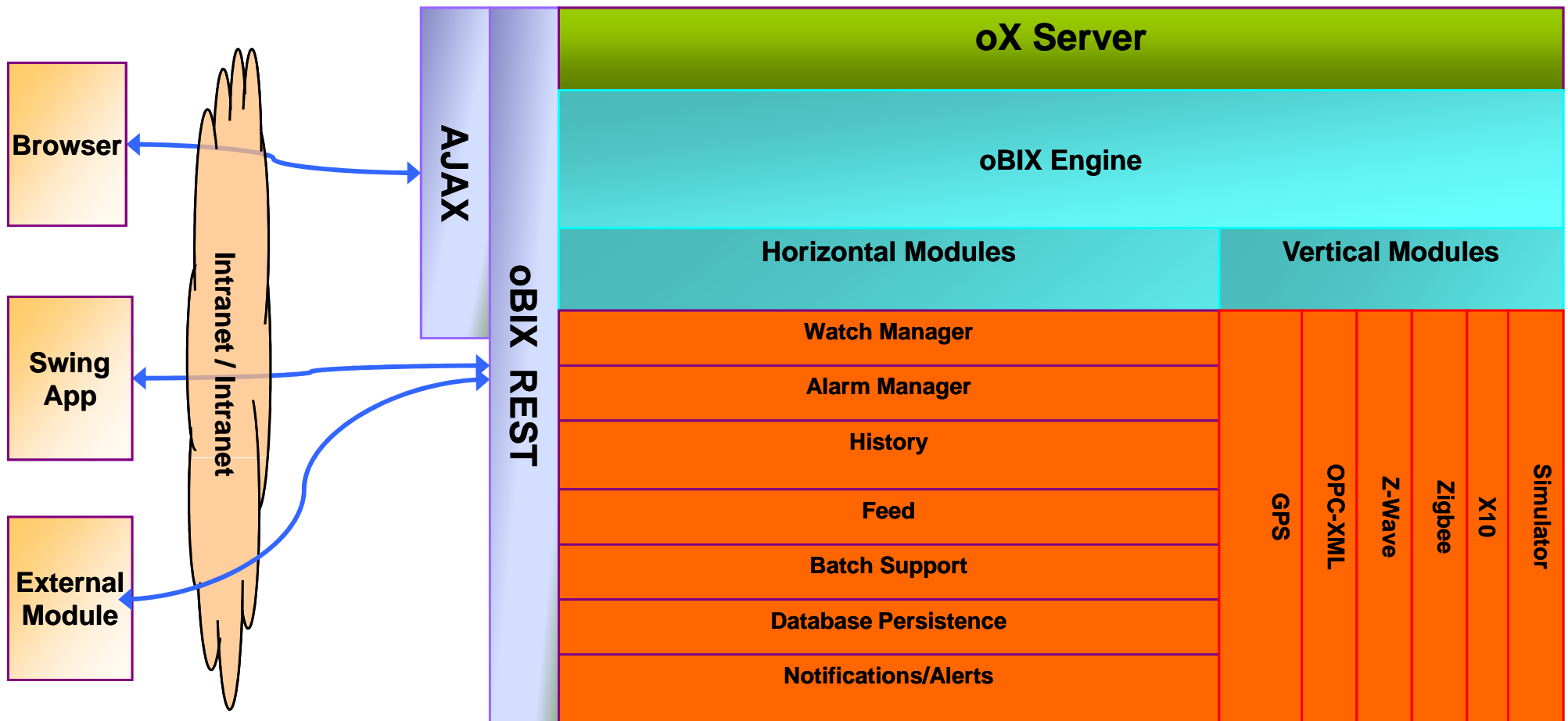


oX Architecture Objectives

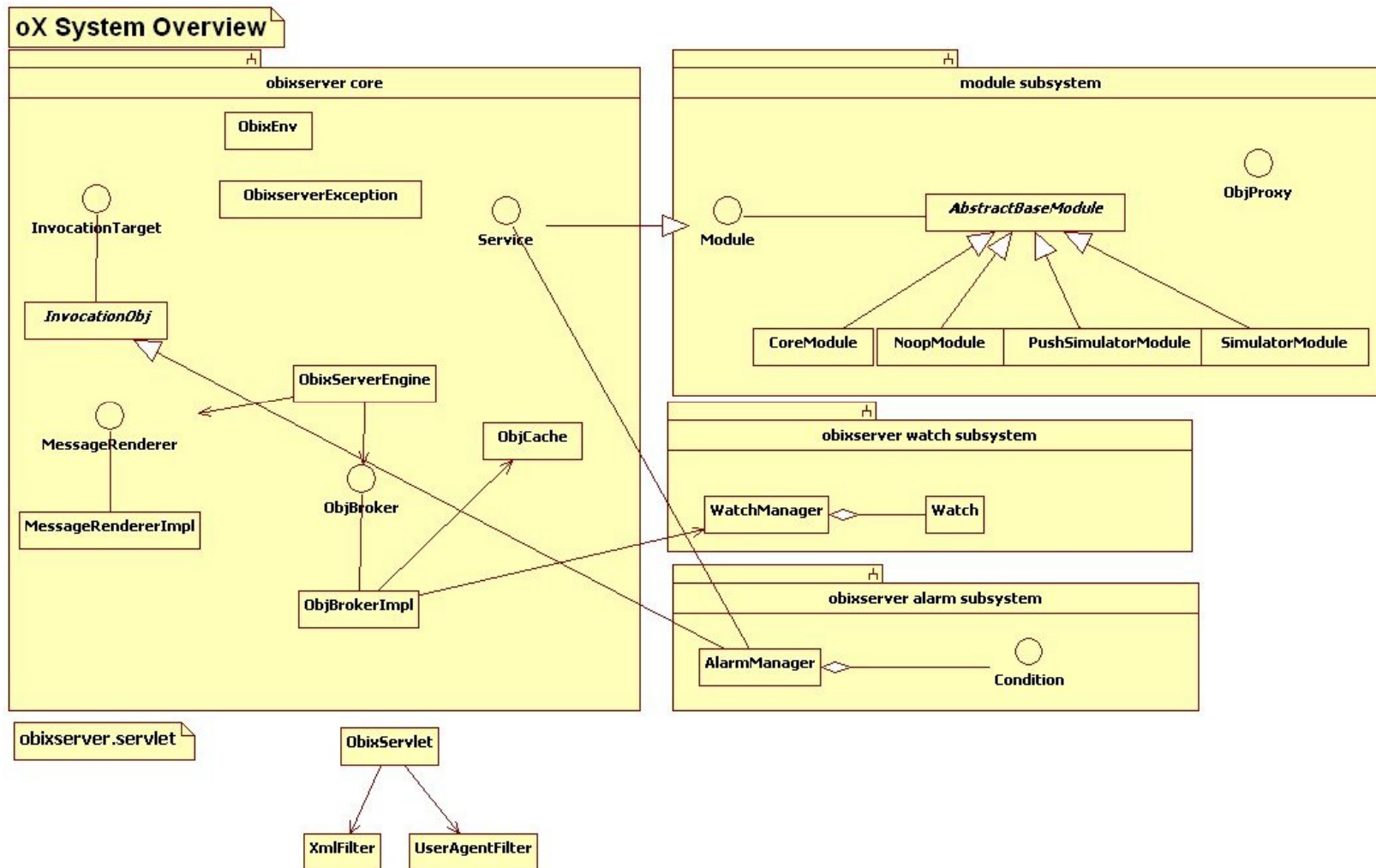
- Integration in projects with "systems which sense"
- Bridging and interop of WSN, pervasive computing systems, Zigbee/802.15.4, X10, Z-Wave, ...
- Integration into Enterprise SOA, internet mashups
- Hosting on low-power device
- Approach
 - Light-weight, runs on J2ME
 - Secure
 - Scalable
 - Extensible
 - » Plugin modules

How oX Works

oX 0.2 Architecture



High Level Class Diagram



oBIX Server Implementation

– Implementation

- Java selected as the most common language and because core oBIX implementation already exists in Java
- Efficient:
 - Currently tested on:
 - » A system with 256MB RAM with 20+K objects running

Plugin Modules

- Internal Modules
 - Higher performance (no REST call overhead)
 - In-process
 - Written in Java
- External Modules
 - Ability to execute remotely via REST
 - Support for any language (C, Python, Perl, ...)
 - Security enforced by standard methods:
 - HTTPS/TLS
 - Basic Authentication

Security

- Standard Security:
 - Important for any industrial or home network deployment
 - Leveraging standard tools that come with Java servlet containers:
 - HTTPS, Basic Authentication
- Extended Security Features
 - Can be added: acegi, ACL based security etc.
 - These are not on oX product roadmap right now, but would like to hear feedback

Footprint/Performance

- Size matters
 - Footprint has to be small to fit on resource-constrained devices
 - Minimum configuration footprint: 500K
- Performance matters
 - Currently supports between 10 and 100 req/s depending on context and server characteristics (lower side on low-level servers like Cappuccino mini PC)

Future

- Complete Implementation of oBIX basic features
 - Batch
 - Query for feed/history
- Implement modules and bridges
 - Zigbee/Zwave/X10
 - Atom/RSS
 - OPC-XML
 - Others, domain specific
- Client and Testing Tools:
 - Simple AJAX charting and monitoring
- Longer term:
 - Suggestions? What would you like to see?

Links/More Information

- Links:
 - <http://obix-server.sourceforge.net>
 - <http://obix.sourceforge.net>
- Articles
 - <http://www.automatedbuildings.com/news/jan05/interviews/hansen.htm>
 - <http://www.newdaedalus.com/articles/>
 - "Brian Frank" Interesting XML ideas in oBIX:
 - <http://lists.xml.org/archives/xml-dev/200701/msg00164.html>
 - Niagara-central – tutorials, e.g.:
 - <http://niagara-central.com/ord?portal:/blog/BlogEntry/112>

Call to Action

- Download and take it for a test ride:
 - <http://obix-server.sourceforge.net>
 - oX user guide in PDF, html formats, including 'How to write external and internal modules'
- Participate in obix-developers group:
 - <http://groups.google.com/group/obix-developers>
- Looking for module implementers
- Looking for module testers

Q/A