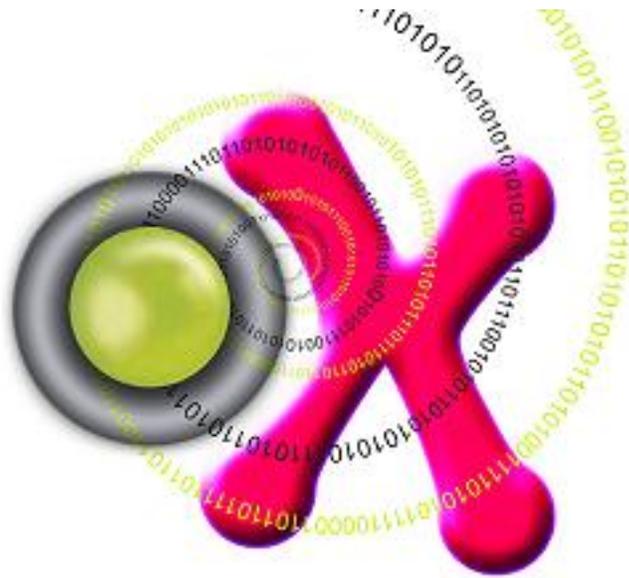


oX White Paper

Peter Michalek



V 0.3

Copyright © 2012 Peter Michalek

Table of Contents

1. Background	1
2. Architecture	2
2.1. Performance Goals	3
2.2. Overview of Architecture	3
2.3. Implementation Language	4
3. Ontologies	5
4. oX License	6

1. Background

Pervasive computing is the next wave of computing innovations that pushes computing into our homes, gadgets we use and into our environment. It has accelerated in the last decade. It manifests itself via different technologies and is called different names, including:

- Internet of Things
- Mobile Computing
- M2M (machine to machine communications)
- Wireless Sensor Networks
- Etc.

Here are some of the factors driving pervasive computing:

- Gradually improving affordability of sensing hardware
- Energy crisis and the need for a universal grid connecting energy supplies and users with buildings, factories and the environment
- Environmental movement and striving for cleaner environment
- Etc.

oX framework proposes a scalable and comprehensive conceptual solution to support the needs of this next wave of computing.

It also outlines how this conceptual solution can be implemented and provides a set of core components as a reference implementation.

oX as a concept can be re-applied and customized to different areas. For example, the OpenLynx project that defines smart grid and energy management monitoring and integration in the context of buildings management is one such customization.

2. Architecture

Here is the overview of the architecture.

The following aspects and features are targeted in the architecture

- Core communications protocol and execution engine
- Modular architecture with Support for Arbitrary Language
- High Degree of Cofigurability
- Option of Distributed Deployment
- Support of resource-constrained machines, such as wireless sensor platforms and mobile devices
- Internal Message Bus
- External Messaging
- Enterprise Application Integration (Interfacing with IT systems)
- Support for Additional Protocols
- Support for Multiple Languages in User Interfaces

oX 0.3 – Framework Architecture

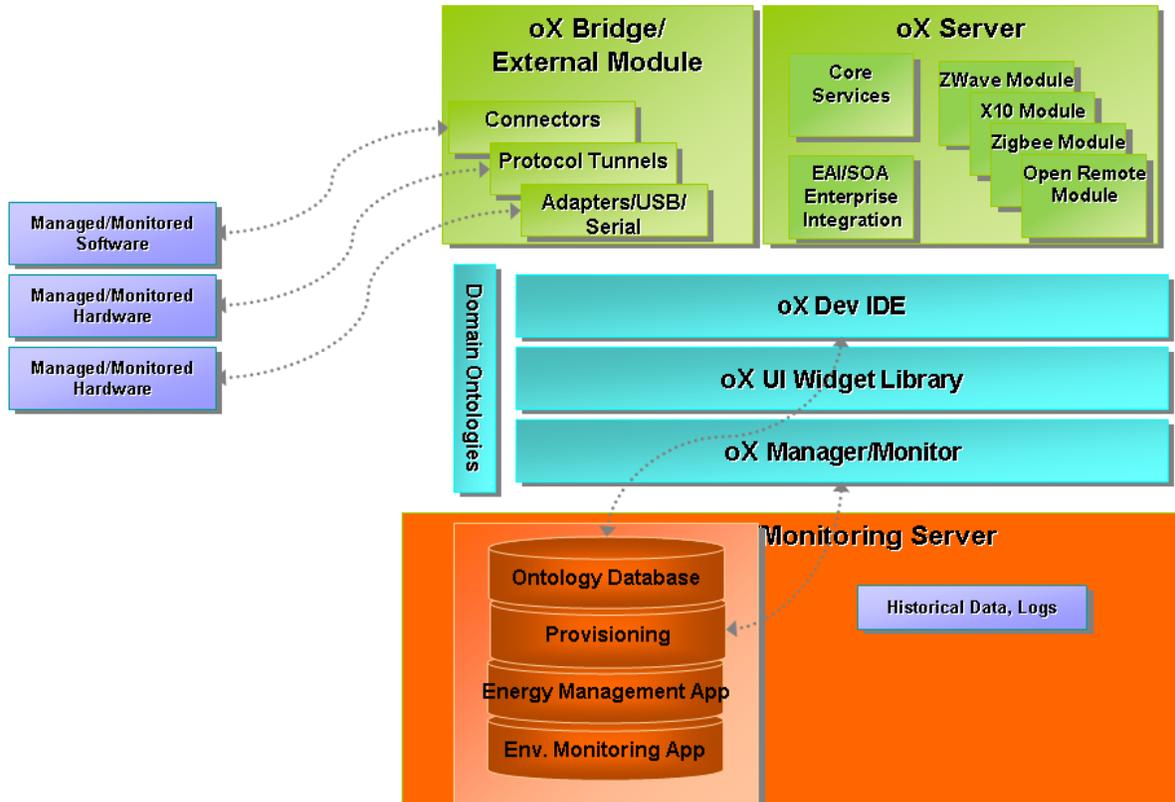


Figure 2.1. High Level Architecture of oX

At high-level, oX framework consists of oBIX server, managed and monitored hardware, IDE and tools and libraries and database.

Horizontal modules provide essential support for oBIX functionality, such as Watch, Alarm, Feed and History.

2.1 Performance Goals

- Millisecond latency when using optimized protocols
- < 50ms latency with REST and Web Services
- Modular architecture allows for very small footprint for special deployments, such as several KB

The system will be architected so that component re-writes or major modifications won't be needed when the system needs to be scaled to hundreds of simultaneous users or thousands of devices.

2.2 Overview of Architecture

A comprehensive architecture needs to be developed that offers components, libraries, tools and an IDE that facilitates configuration and deployment of the components and development of new components, automated updates of subsystems, semi-automated or automated functional and security tests and validations, etc.

Since this is a fairly ambitious objective, it will have to be phased in, with the key components and most important aspects designed and implemented first.

2.3 Implementation Language

Different components of the framework may be developed using different languages, based on the optimal fit for the task and the preferences of the project team.

This may include, among others, the following languages:

- Java
- Python
- Lisp
- Lua
- Ruby on Rails
- PHP
- C/C++
- Groovy/Grails
- .NET languages (C#, VB.NET, etc.)
- Innovative new languages such as Fan (<http://fandev.org/>), Clojure, etc.
- AJAX/JavaScript
- Flex/ActionScript
- Objective C

The recommended version of the language used is the latest available, unless a lower version is required because of component dependencies or low-resource platform requirements. Note that the last few items of the list above reflect the need to support languages that are used in Web based RIA (Rich Internet Applications) client interfaces and on mobile devices such as iPhone and Android-based smart phones.

3. Ontologies

oX framework can be applied to many domains that deal with systems that sense or systems that need to be monitored. Ontologies play an important role in the oX framework to enable rapid application development in these domains. They allow a simple way of expressing the needs for storage, communications and integration in the area to which a particular oX application is applied.

According to the wikipedia entry on ontology:

Ontology (information science) [[http://en.wikipedia.org/wiki/Ontology_\(computer_science\)](http://en.wikipedia.org/wiki/Ontology_(computer_science))]:

“In computer science and information science, an ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain, and may be used to define the domain.”

More specifically, within the context of the Semantic Web and oX, we understand ontology as a form of knowledge representation about a particular domain that an oX based software deployment is addressing. It can address it by collecting data from devices and sensors, performing actions on the software and on the environment. These actions are driven via events and alarms or they can result from user and system commands on the actuators of devices that may result in physical changes in the environment.

A particular oX deployment may define a nominal ontology for a particular domain, such as home automation, automated metering, patient care, asset and merchandise tracking or environmental monitoring.

It may also define additional derived ontologies for integration with systems addressing the same domain or a similar domain or with generic IT systems.

In oX framework, ontologies are defined in terms of the underlying oBIX system - we use oBIX contracts to define them.

4. oX License

oX framework license is academic MIT-style. It allows free use, distribution and extension of the software.