Challenges with OpenXT Messaging

- Messaging with OpenXT actually uses two different patterns
  - RPC invocation to request an action or for information from a remote subsystem
  - Pub/Sub messaging for near-realtime event notification
- Need to support for v4v transport
- Need to be flexible to be adapted to be an SELinux User Space Object Manager

IDL RPC Solutions

- IDL solutions are popular as they often have binding generators for multiple languages
- Popular IDLs are Thrift, GRPC, Cap'n Proto, Avro IDL
- These work great for solving the RPC invocation
- To make eventing work, two approach are possible
  - Polling (ugh)
  - Callback (now client has to act as a server)

WAMP

- WAMP is an open standard protocol that implements RPC and Pub/Sub
  - It is transport and marshaling agnostic, default being WebSocket and JSON
  - A registered WebSocket protocol with IANA
  - Focus is on IoT solutions, therefore overhead is an important consideration
  - Basis is built around the “Router” which is a combination of a RPC Dealer and Pub/Sub Broker

OpenXT WAMP Components

- Security Enhanced Router
  - A User Space Object Manager for SELinux
  - Leverages WAMP Realm, Session, and Peer concepts
  - Leverages upcoming Argo for session labeling
  - Based on Nexus WAMP router
    - Written in pure Go
    - Provides single static binary

OpenXT WAMP Flow

- A single instance of an SERouter will function as name server
  - Discovered in XenStore (or alternatively via static/well known domain UUID)
  - Will be able to enforce SELinux policy for service look up calls
- An SERouter contains a proxy client
  - When a local RPC or topic is published, proxy will register with name server
  - When a local call comes in
    - if not registered local, proxy will contact name server
    - proxy will establish direct connection to remote vm's router
    - proxy will mediate call and response to/from remote VM's router
  - When a local subscription comes in
    - if not registered local, proxy will contact name server
    - proxy will establish direct connection to remote vm's router
    - proxy will register subscription and mediate any incoming messages