

California Demand Response Business Network (DRBizNet)

Architecture Overview

- a work in progress -

Scott Neumann
Utility Integration Solutions, Inc.

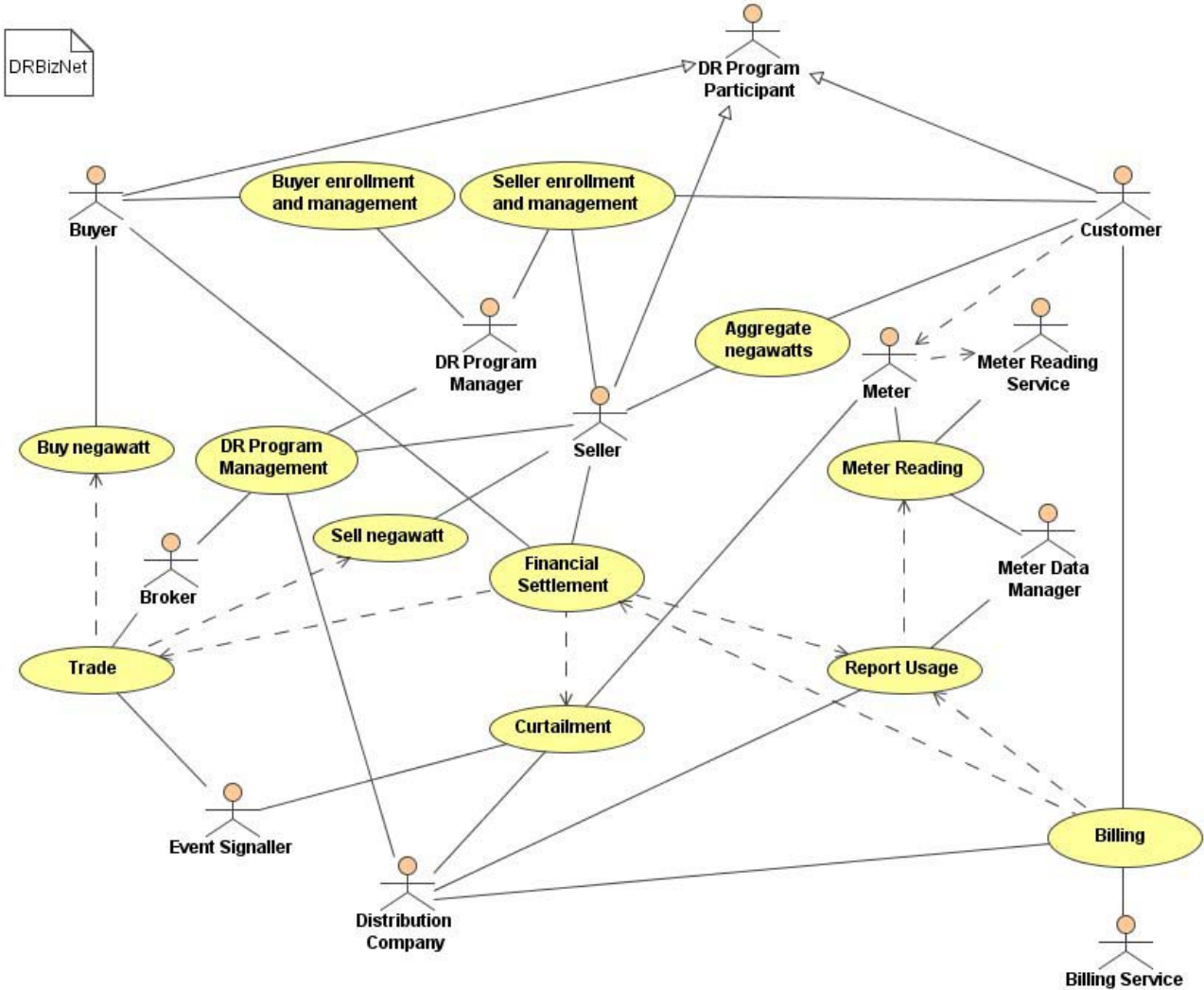
Eamonn McCormick
Dynamic Networks

June 10, 2004

Introduction

- The purpose of this presentation is to provide an overview of the DRBizNet architecture.
- This is a work in progress. Several draft documents related to architecture and technologies are on the DR Project Portal
- This presentation is an early draft of the design, covering:
 - High-level modeling
 - Relevant standards
 - Applicable technologies
 - Components and deployment

DR Use Case Overview



Participants

- Buyers
- Sellers
- Program Manager
- Meter Service Providers
- Billing Service Provider
- Local Distribution Company
- Broker
- Customer (customers may be aggregated by a seller known as an Aggregator, or some customers may sell directly)

Participants can take on one or more roles, and new roles can be added as needed!

Roles

- Program Manager
- Distribution Company
- Regulator
- Market Monitor
- Buyer
- Seller
- Aggregator
- Broker
- Meter Reader
- Meter Data Manager
- Billing Agent
- Settlement Agent
- ... *others can be defined as needed*

Key DR Use Cases We Focused On

- Buyer enrollment and management
- Seller enrollment and management
- Program management
- Trading
- Curtailment
- Settlements
- Aggregation

Categories of DR Business Services

- Buyer enrollment
- Seller enrollment
- Create program
- Manage program
- Trading
- Metering
- Settlements
- Billing and Payment

Potential DR Business Services

- Accept Aggregate Price Curve Buy Reply
- Activate Network Participant
- Activate Network Participant User
- Activate Trading Partner Relationship
- Create Aggregate Price Curve
- Create Aggregate Price Curve Buy Reply
- Create Capacity Buy Request
- Create Capacity Buy Request Relationship
- Create Capacity Buy Request Relationship Name
- Create Capacity Buy Request Type
- Create Load Relationship
- Create Load Relationship Name
- Create Load Type
- Create Location
- Create Location Relationship
- Create Location Relationship Name
- Create Location Type
- Create Program
- Create Program Type
- Create Network Participant
- Create Network Participant User
- Create Actor Role Location Relationship Name
- Create Actor Role Relationship Name
- Create Price Curve Bid
- Create Price Curve Characteristic
- Create Role Configuration
- Create Role Relationship Relationship Name
- Create Trading Partner Relationship
- Forward Aggregate Price Curve
- Forward Aggregate Price Curve Buy Reply
- Forward Capacity Buy Request
- Get Aggregate Price Curve
- Get Aggregate Price Curve Buy Reply
- Get Capacity Buy Request
- Get Capacity Buy Request Relationship
- Get Capacity Buy Request Relationship Name
- Get Capacity Buy Request Type
- Get Load Relationship
- Get Load Relationship Name
- Get Load Type
- Get Location
- Get Location Relationship
- Get Location Relationship Name
- Get Location Type
- Get Program
- Get Program Type
- Get Network Participant
- Get Network Participant User
- Get Actor Role Location Relationship Name
- Get Actor Role Relationship Name
- Get Price Curve Bid
- Get Price Curve Characteristic
- Get Role Configuration
- Get Role Relationship Relationship Name
- Get Trading Partner Relationship
- List Aggregate Price Curves
- List Aggregate Price Curve Buy Replies
- List Capacity Buy Requests
- List Capacity Buy Request Relationships
- List Capacity Buy Request Relationship Names
- List Capacity Buy Request Types
- List Load Relationships
- List Load Relationship Names
- List Load Types

Potential DR Business Services (continued)

- List Locations
- List Location Relationships
- List Location Relationship Names
- List Location Types
- List Programs
- List Program Types
- List Network Participants
- List Network Participant Users
- List Actor Role Location Relationship Names
- List Actor Role Relationship Names
- List Price Curve Bids
- List Price Curve Characteristics
- List Role Configurations
- List Role Relationship Relationship Names
- List Trading Partner Relationships
- Maintain Network Participant Contacts
- Maintain Network Participant Information
- Maintain Network Participant Location Contacts
- Match Capacity Buy Request to Price Curve Bid
- Publish Network Participant Updates
- Receive Aggregate Price Curve
- Receive Aggregate Price Curve Buy Reply
- Receive Capacity Buy Request
- Receive Forwarded Capacity Buy Request
- Receive Forwarded Aggregate Price Curve Buy Reply
- Receive Price Curve Bid
- Reject Price Curve Bid
- Remove Capacity Buy Request from Program
- Remove Price Curve Bid
- Renew Capacity Buy Request
- Retrieve Network Participant Directory Information
- Schedule Hour Ahead Capacity Requirement
- Select Capacity Buy Request Recipients
- Select Qualified Service Providers
- Send Aggregate Price Curve Buy Reply
- Send Capacity Buy Request
- Send Price Curve Bid
- Update Capacity Buy Request Relationship Name
- Update Capacity Buy Request Type
- Update Load Relationship Name
- Update Load Type
- Update Location
- Update Location Relationship
- Update Location Relationship Name
- Update Location Type
- Update Program
- Update Program Type
- Update Network Participant
- Update Network Participant User
- Update Actor Location Relationship Name
- Update Actor Relationship Name
- Update Price Curve Characteristic
- Update Actor Configuration
- Update Actor Relationship Relationship Name
- Update Trading Partner Relationship
- Validate Aggregate Price Curve
- Validate Aggregate Price Curve Buy Reply
- Validate Capacity Buy Request
- Validate Forwarded Capacity Buy Request
- Validate Forwarded Aggregate Price Curve Buy Reply
- Validate Price Curve Bid

DR Business Message Simplifications:

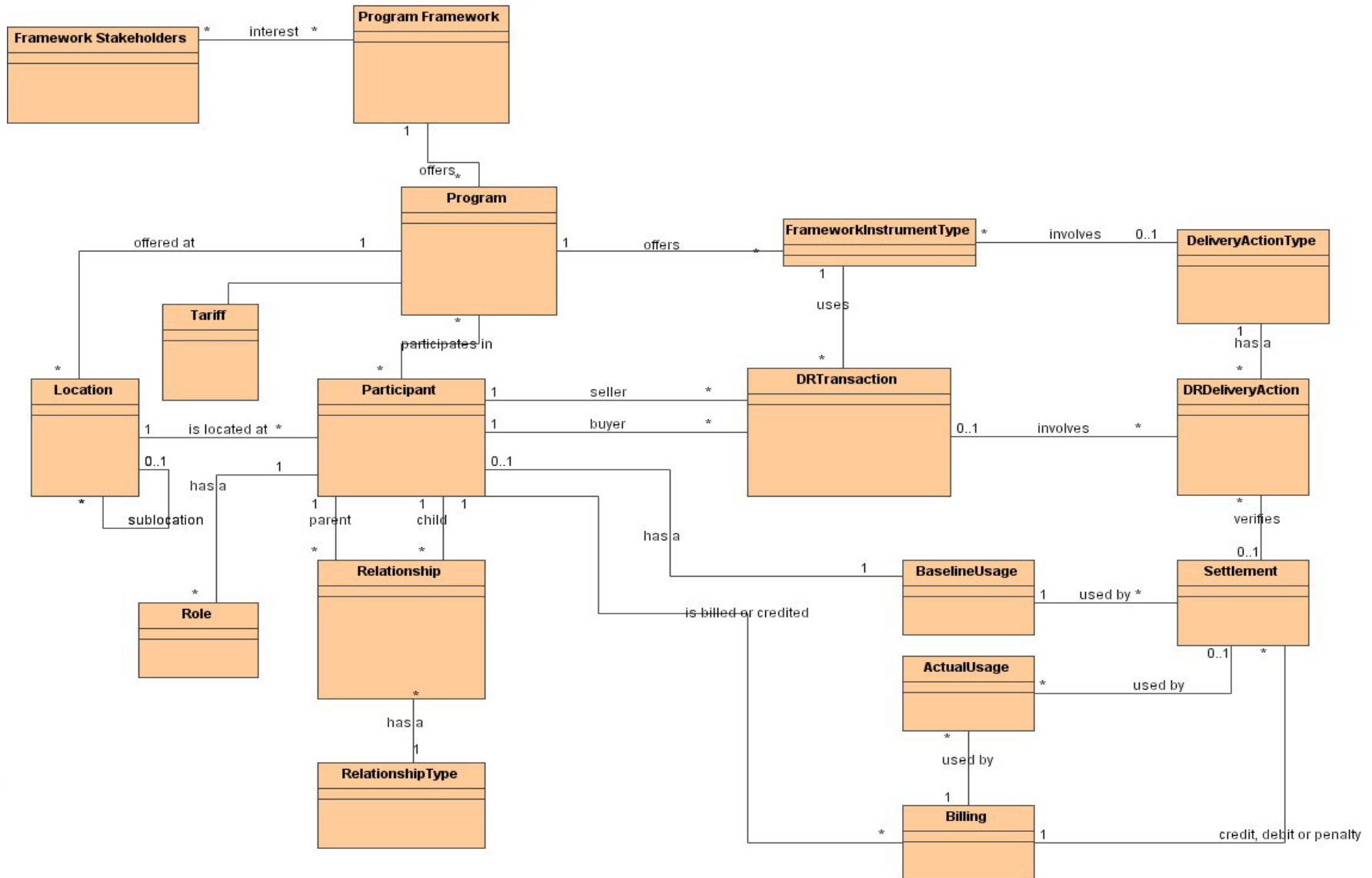
Verbs:

- Create
- Get
- Show
- Update
- Cancel
- Delete
- Validate

Nouns:

- Participant
- Price Curve
- Buy Request
- Load
- Location
- Program
- Role
- Relationship

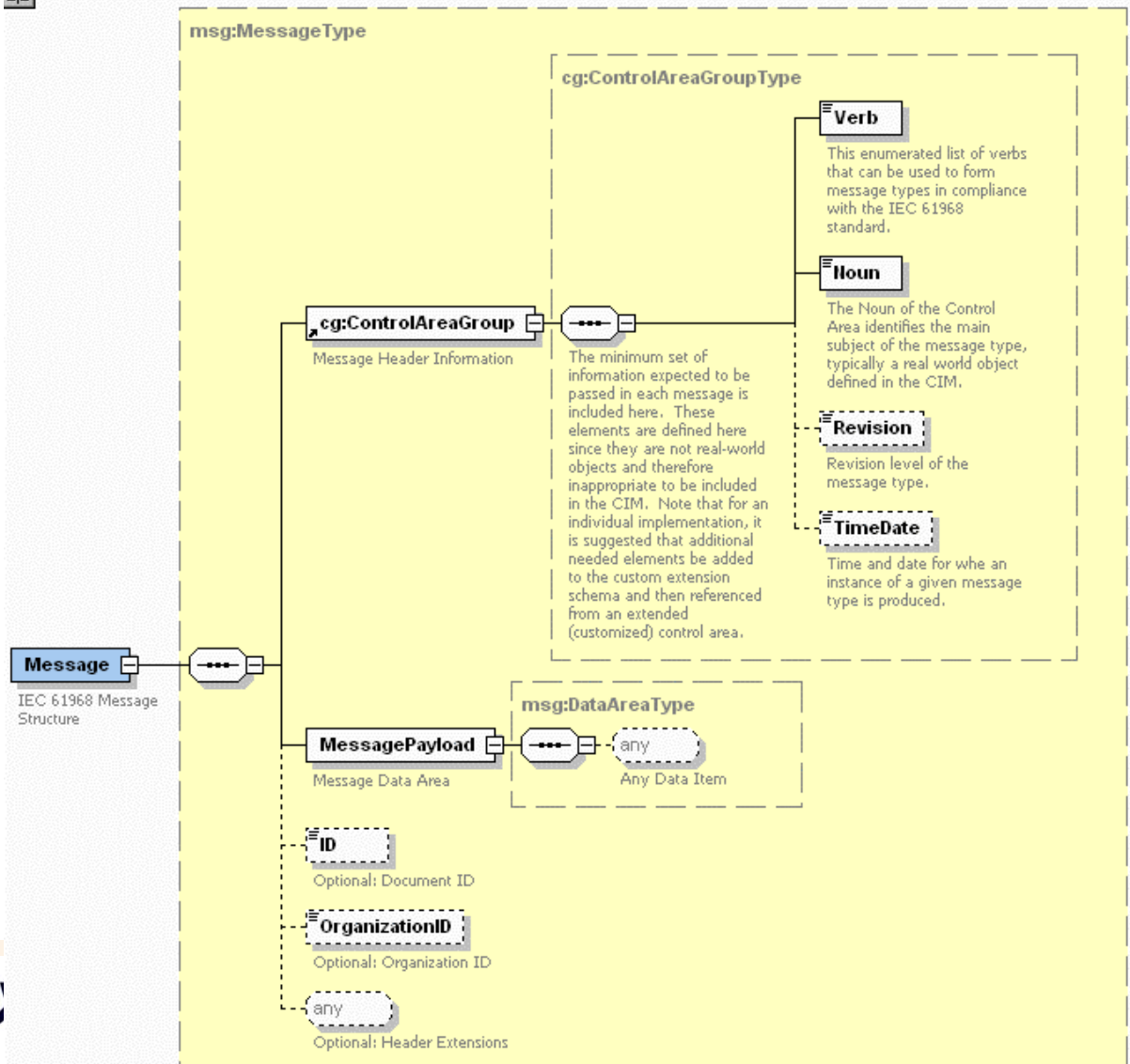
Logical Information Model



Semantics

- A *Framework* can have a set of *Programs*
- A *Program* has a *Tariff* and a *FrameworkInstrumentType*
- *Participants* can be: buys, sellers, service providers
- *Participants* can be associated to *Programs* and assume one or more *Roles*
- *Participants* can be associated to other *Participants* through *Relationships*
- *Participants* and *Programs* are associated with *Locations*
- A *Transaction* can occur between a buyer and a seller for a *FrameworkInstrument*
- A *Transaction* may involve a *DeliveryAction*
- A *Settlement* may involve the verification of a *DeliveryAction* and impact on Usage

Generic Message Structure



Generic Message Structures

- Leverage XML and web services
- Support a few generalized, flexible interfaces instead of many very specific, inflexible interfaces
- Use noun, verb and payload combinations, where the verb identifies the type of action and the noun identifies the type of the payload, which usually related to a business document
- This approach is consistent with (and improves upon) IEC 61968
- Use of verb-noun combinations for messaging with supporting software design can often reduce the order of complexity from an $M \times N$ problem to more that of an $M + N$ problem.
- Genericity and flexibility are key to supporting an ongoing evolution of DR programs

Key Technology Trends

- Use of the internet for business
- New generation of standards to support more flexible integration, e.g. XML, web services, Java, ...
- Service Oriented Architectures (SOA)
- Wide range of open standards and associated products
- Distribution of processing
- Wide range of low cost mechanisms for internet integration
- Business processes defined as an orchestration of web services (e.g. BPEL)

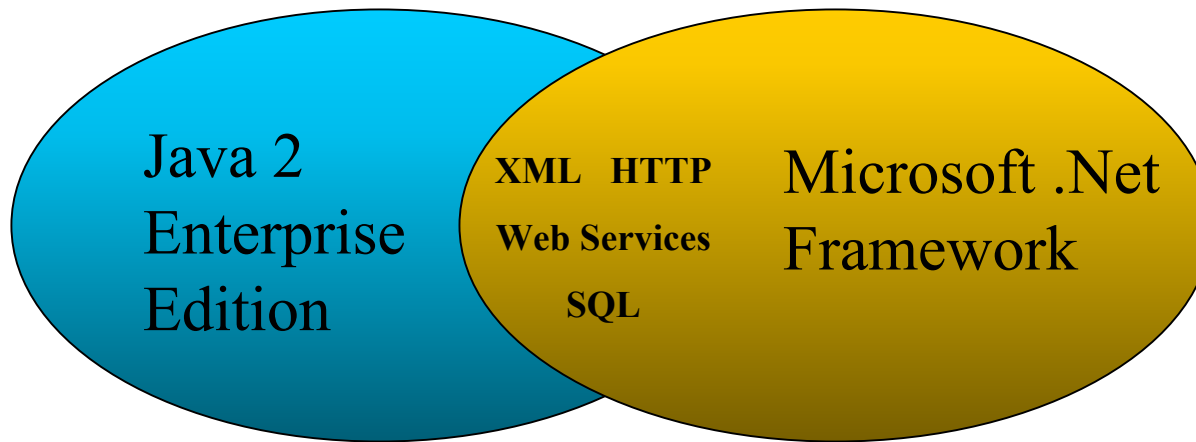
Applicable Industry Standards

- EPRI/IEC CIM
- IEC 61968
- OPC XML
- Utility Industry Group (UIG)
- ANSI C12
- ANSI X12

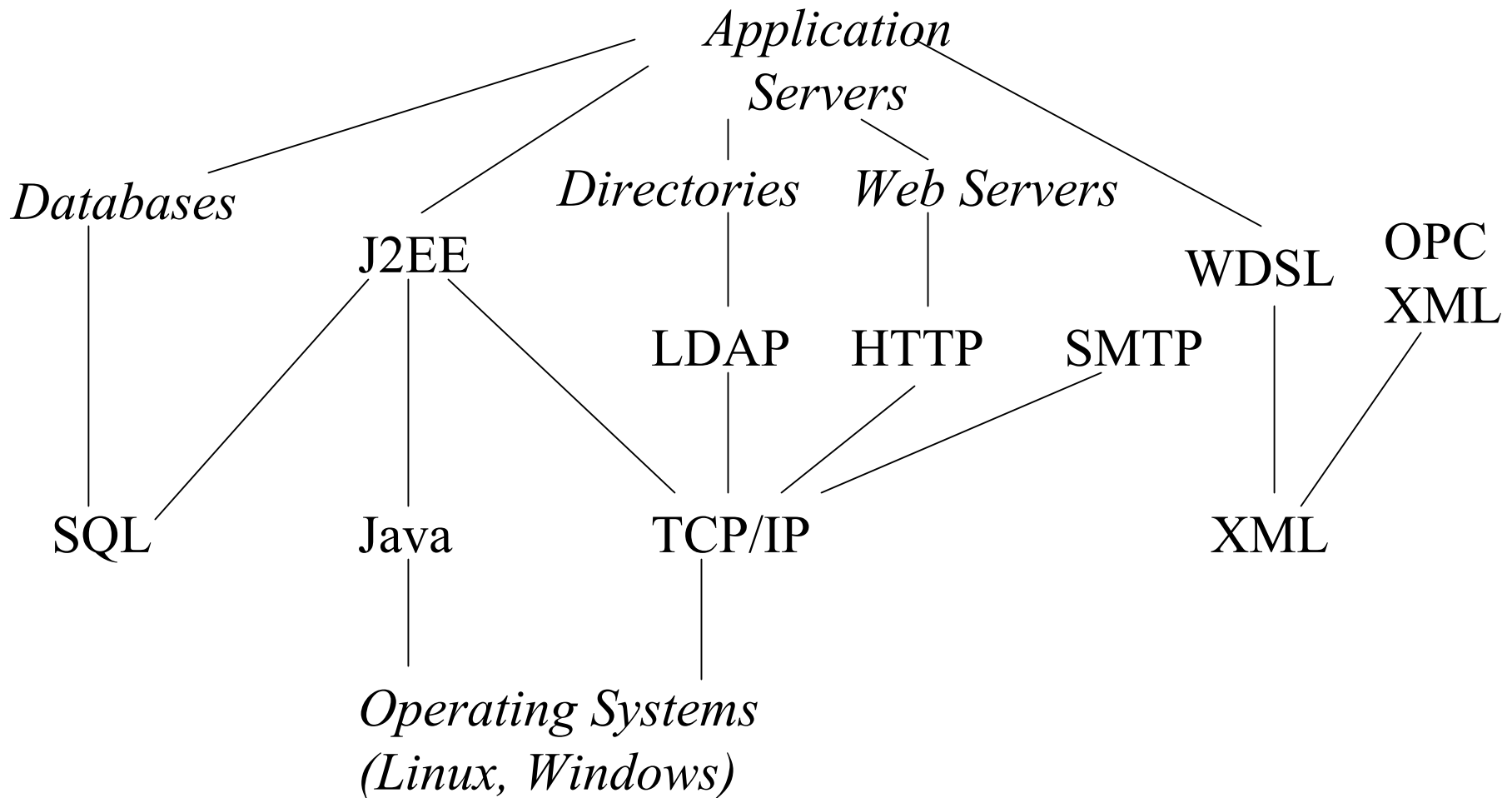
Utility Industry Group

- UIG has defined transactions for the following information flows:
 - 814 - Enrollment and Maintenance
 - 650 - Meter Information
 - 867 - Meter Readings
 - 810 - Billing
 - 568 - Collected Remittances
 - 820 - Payment
 - 248 - Write-Off
 - 824 - Application Advice
- These transactions could/should be applied as appropriate to DRBizNet

Software Architectures and Standards



Technologies and Dependencies



Java 2 Enterprise Edition (J2EE)

- Supports enterprise capabilities for Java applications
- Provides Java Database Connectivity (JDBC) for database integration
- Provides Java Messaging System (JMS) for messaging
- Application server and integration products are most frequently based upon J2EE
- Through the use of web services and XML, integration with non-J2EE environments, such as Microsoft .Net is transparent

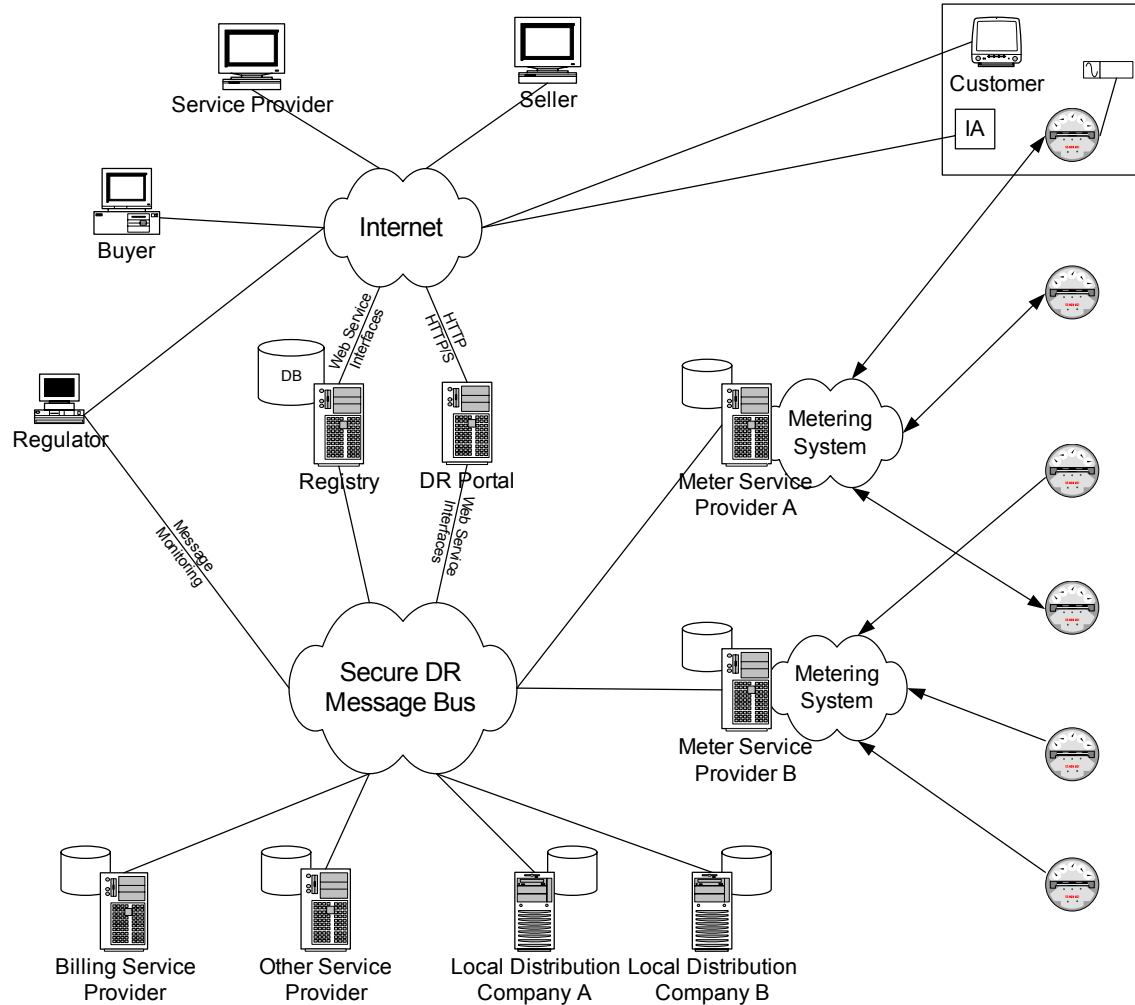
Open Source Products

- Open source products often mirror commercial products and provide options for low cost, standards-based deployments of systems
- Examples of open source products:
 - Operating System: Linux
 - Database: MySQL
 - Web Server: Apache
 - Application Server: JBoss
- Product vendors now actively support, leverage and certify open source products

DRBizNet Components

- DR Registry: To manage participants, relationships and transactions
- DR Web Services: For access of business services
- DR Message Bus: Used for communication between components
- DR Connector: To permit secure B2B connections to the DR message bus
- DR Portal: To permit human access to the DR framework and associated web services
- Intelligent Agents: Can make decisions on behalf of a participant with configurable rules without human intervention. Could be deployed on a server, PC or within an end device

DRBizNet Deployment



DR Registry

- Manages definitions of programs, participants and the relationships between participants
- Manages and tracks DR transactions
- Publishes notifications on message bus
- Could be implemented as a single, centralized service or as a distributed service (similar to DNS or LDAP)
- Constructed using an application server and a database

DR Web Services

- DR business functions are made available to participants using web services
- Web services are defined using WSDL and XML Schema in a generalized manner as previously described
- Business logic is implemented using a J2EE-based application server
- Web services can be invoked directly by service provider applications or through a DR portal by participants

DR Message Bus

- Provides for real-time communications between service providers and associated systems
- Provides for the orchestration of DR business processes (might decide to use BPEL)
- Use of publish/subscribe bus can permit effective market monitoring
- Issues notifications using a variety of mechanisms, including web services and e-mail

DR Connector

- Provides the means for service providers to connect to the DR Message Bus in a secure, reliable manner
- Provides the means to send and receive encrypted XML messages
- Will typically utilize the internet for connectivity
- Provides necessary delivery guarantees

Intelligent Agents

- Participants may want and should the ability to make DR decisions automatically in real-time without the need for human intervention
- Intelligent agents are low cost software components that can make decisions on the behalf of participants in real-time based on pre-defined rules
- Have interfaces (e.g. web services) that can be readily invoked to initiate processing, messages would be XML-based as described previously
- Intelligent agents can be configured to execute transactions, control devices (e.g. invoke delivery actions) and issue notifications based on defined rules
- Rules can be edited or loaded dynamically
- Can be physically embedded or installed on participant devices, PCs or servers

Design Criteria

- **Flexibility**: Use of generalized message structures based on IEC 61968
- **Cost Minimization**: Architecture provides the ability to leverage low cost solutions
- **Speed of Operation**: Enhanced by the distribution of processing
- **Interoperability**: Achieved through use of standards based upon XML and web services
- **Reliability**: Will be improved through loose coupling, generalization of components and minimization of custom code
- **Security**: To be provided by use of DR Connector and adherence to appropriate security standards
- **Scalability**: Supported by use of a message bus which in turn permits distribution and federation of processing

Summary

- This architecture is a work in progress, which has started utilizing drafts documents developed to date
- A prototype will be developed to demonstrate the architecture
- Prototype will focus on several key use cases
- Will need to finalize a variety of technical decisions for implementation of the prototype