The Enterprise Backbone (EBb)

Distribution of Services, Events and Data

An Information Delivery Engineering Viewpoint

Protocol Design

Please contribute {8>D)

Contents

Please contribute {8>D)

Overview

Motivation

Intended Use

The
Information Delivery
Engineering Viewpoint

Please contribute {8>D)

Motivation

Protocol Design

For an Information Delivery IT Strategy, answer three questions from an Engineering Viewpoint

- What is the effective set of service, event, and data distribution protocols necessary to provide a fully functional Enterprise Backbone?
- What components are needed to implement an Enterprise Backbone?
- What are some design patterns to assist in the implementation, deployment and use of the proposed EBb protocols?

Please contribute {8>D)

Intended Use

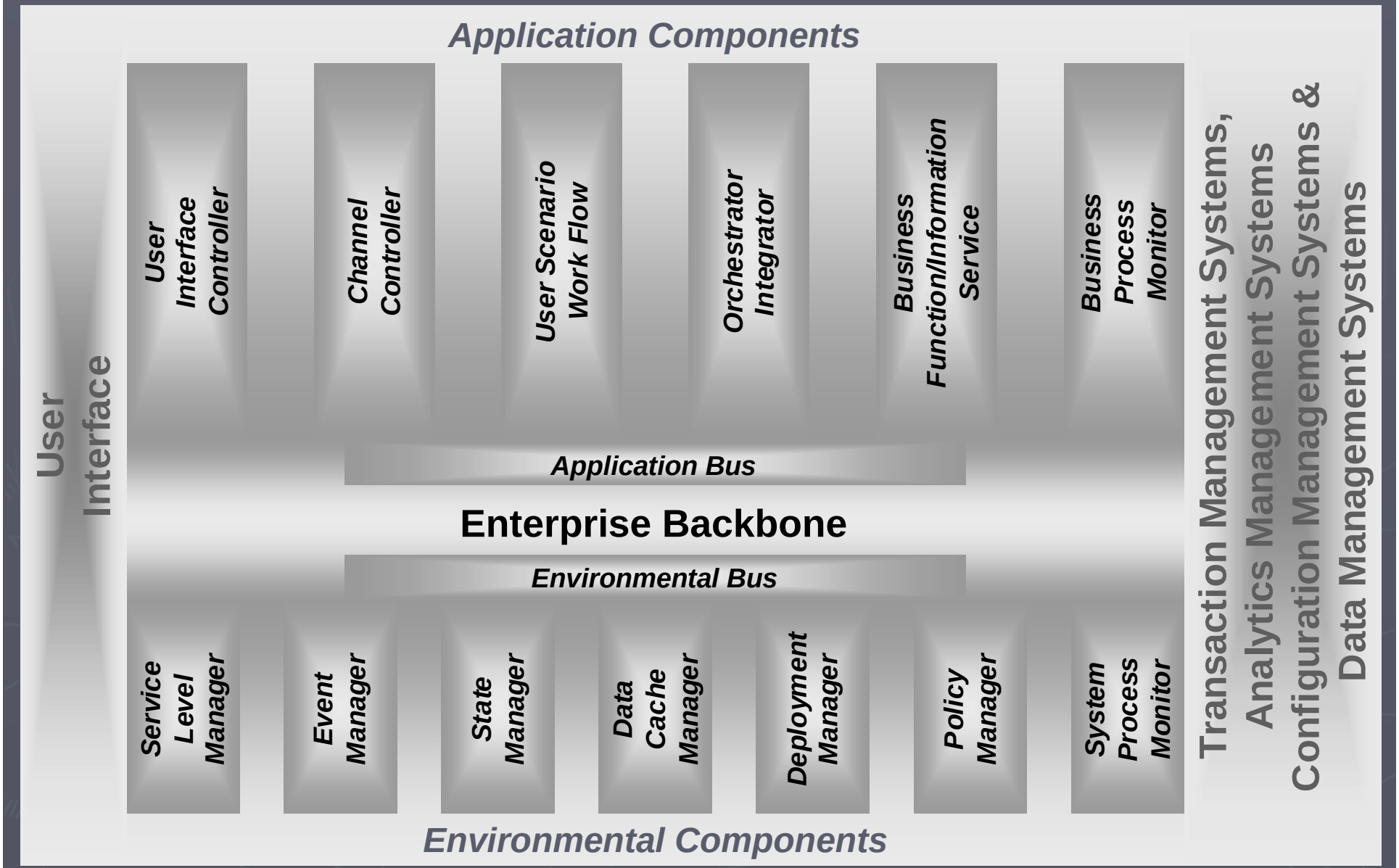
Protocol Design

- Enterprise Architecture Statement
  - Target Architecture of Record
- Evaluation Framework
  - Tools
  - Methods
  - Technology
- Guide for Application Development
  - Implement SOA COE Strategy
  - Implement Data Management Transformation Strategy

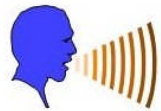
Please contribute {8>D)

The Information Delivery Engineering Viewpoint

Protocol Design



Please contribute {8>D)



Base Concepts

Event

Topic

View

Bus/Conduit

YellowBoard

Please contribute {8>D)

Base Concepts
Primitive Logical System Resources

Protocol Design

Event
- A significant occurrence at the Business Process, Application Execution or System Operation level that is identified in a Registry

Topic
- A categorical key word or key phrase (possible sequence of either) that is defined within a taxonomic or semantic structure

View
- Specification of a data context needed to support the realization of an Application Scenario or execution of a Service
  - Can be articulated by an SQL statement)

Bus
- A collection of conduits through which flow service requests, events and/or data:
  - Application
    - Supports collaborations amongst components to realize a specific User Experience
  - Environmental
    - Supports collaborations amongst components used to control operations of systems

YellowBoard
- A global system structure that allows posting and reading of available resources and states of services in flight

Please contribute {8>D)

Supporting Components for Base Concepts

Protocol Design

YellowBoard

Resource & Service Status Postings

Event Registry

Topic Registry

View Registry

conduits ***Application Bus***

Enterprise Backbone

conduits ***Environmental Bus***

Please contribute {8>D)

EBb Protocols

Common Functionality

Reliable Broadcast

Publish Subscribe

Request Reply

Data Stream

Please contribute {8>D)

Common EBb Protocol Functionality

Protocol Design

Service Points

- advertiseService({ (aServiceDescription, aVoI, aServiceLevelOffer) }, aYellowBoard, aProvider) returns (aPostingRecord)
  - Advertise on a YellowBoard, a set of service and level offerings { (**S**, **VoI, SLO**)$_i$ } by a provider
- locateService(aServiceDescription, aServiceLevelRequirement, aYellowBoard) returns (anSLARecord)
  - Locate on a YellowBoard, a provider of a service described

Usage

- All returns are described as XML documents
  - Can be "compiled" or rendered into more efficient forms for execution
- All System Components have three (wait! four) common instrumentation methods
  Communicated via a designated ***conduit*** on the ***System Management Environmental Bus***
  - reportNameVersion()
    What's your name?
  - reportRequirements()
    What resources do you need to work?
  - reportProtocolList()
    What do you speak?
  - reportUseStatistics()
    What have you done?

Notes

Please contribute {8>D)

Reliable Broadcast Protocol

Service where receiver can know if message/event missed and request resend

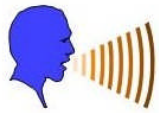
Please contribute {8>D)

Reliable Broadcast Protocol
Service Points

Protocol Design

- subscribeRBEoI({ ({ ({ anEoI$_k$ }, aToI)$_m$ }, aVoI)$_n$ } , aBroadcastProvider)
  returns (aBroadcastRegistration)
  - Register sets of (sets of (list of Events of Interest under a Topic of Interest) within View a of Interest) to a Broadcast Provider
- receiveAsynchRB(aBroadcastRegistration)
  returns (aMessage)
  - Receive Messages asynchronously on the Conduit of a Broadcast Registration
- resendRB(aBroadcastRegistration, lastGoodMessage)
  returns (aStatus)
  - Resend messages after the last good message received through the Conduit of a Broadcast Registration for Messages
- suspendRB(aBroadcastRegistration)
  returns (aStatus)
  - Suspend receipt through the Conduit of a Broadcast Registration for Messages
- resumeRB(aBroadcastRegistration)
  returns (aStatus)
  - Resume receipt through the Conduit of a Broadcast Registration for Messages
- disengageRB(aBroadcastRegistration)
  returns (aStatus)
  - Disengage from the Conduit of a Broadcast Registration for Messages

Please contribute {8>D)



Publish Subscribe Protocol

Service based on taxonomy of topics/event types of interest where consumer and provider are unknown to each other

Please contribute {8>D)

Publish Subscribe Protocol
Service Points

Protocol Design

- subscribePSEoI({ ({ ({ anEoI$_k$ }, aToI)$_m$ }, aVoI)$_n$ } , anInformationProvider) returns (anInformationSubscription)
  - Register sets of (sets of (list of Events of Interest under a Topic of Interest) within View a of Interest) from an information provider
- registerPSEoI({ ({ ({ anEoI$_k$ }, aToI)$_m$ }, aVoI)$_n$ } , anInformationProvider) returns (anInformationPublication)
  - Register sets of (sets of (list of Events of Interest under a Topic of Interest) within View a of Interest) as an information provider
- receiveAsynchPS(anInformationSubscription) returns (aMessage)
  - Receive Messages asynchronously on the Conduit of a Information Subscription
- publishPSEoI(aMessage, anEoI, aToI, aVoI, anInformationPublication) returns (aStatus)
  - Publish message and its context (EoI, ToI, VoI) on the Conduit of a Information Registration
- suspendPSSubscription(anInformationSubscription) returns (aStatus)
  - Suspend receipt through the Conduit of a Subscription for Messages
- resumePSSubscription(anInformationSubscription) returns (aStatus)
  - Resume receipt through the Conduit of a Subscription for Messages
- disengagePSSubscription(anInformationSubscription) returns (aStatus)
  - Disengage from the Conduit of a Subscription for Messages
- suspendPSPublication(anInformationPublication) returns (aStatus)
  - Suspend receipt on the Conduit of a Publication of Messages
- resumePSPublication(anInformationPublication) returns (aStatus)
  - Resume receipt on the Conduit of a Publicatiof Messages
- disengagePSPublication(anInformationPublication) returns (aStatus)
  - Disengage from the Conduit of a Publication of Messages

Please contribute {8>D)

Request Reply Protocol

Service invocation which is potentially dynamically locatable where result is returned

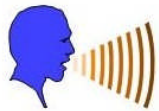
Please contribute {8>D)

Request Reply Protocol
Service Points

Protocol Design

- requestAsynchRR(anSLARecord)
  returns (aConduit)
  - Request asynchronously a Service through a Conduit per an SLA Record
- receiveAsynchRR(aConduit)
  returns (aMessage)
  - Receive Messages asynchronously through the Conduit of a Service request
- requestRR(anSLARecord)
  returns (aMessage)
  - Request synchronously a Service through a Conduit per an SLA Record
- disengageRR(anSLARecord)
  returns (aStatus)
  - Disengage from an asynchronous Service request through a Conduit per an SLA Record

Please contribute {8>D)



Data Stream Protocol

Service that delivers large units of data of differing types, textual or bit-based

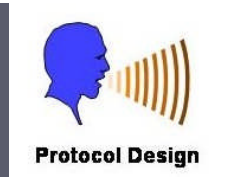
Please contribute {8>D)

Data Stream Protocol
Service Points

Protocol Design

- receiveDS(aConduit, aVoI, aStoreType, aProvider)
  returns (aStoreRecord)
  - Connect and Receive into a store, a data stream of a **VoI** from a provider through a conduit
- suspendDS(aConduit, aVoI, aStoreLocation, aProvider)
  returns (aStopRecord)
  - Suspend receiving a data stream of a **VoI** from a service provider through a conduit
- resumeeDS(aConduit, aVoI, aStoreLocation, aProvider)
  returns (aStopRecord)
  - Restart receiving a data stream of a **VoI** from a service provider through a conduit
- disengageDS(aConduit, aVoI, aStoreLocation, aProvider)
  returns (aStopRecord)
  - Stop receiving a data stream of a **VoI** from a service provider through a conduit

Please contribute {8>D)



EBb Protocol Design Patterns

Application Scenarios

- Data Replication
- Migration Transparency

Please contribute {8>D)

Data Replication

Protocol Design

Please contribute {8>D)

Migration Transparency

Protocol Design

Please contribute {8>D)