



Database Systems

Session 6 – Main Theme Enterprise Reference Architecture Blueprinting

Dr. Jean-Claude Franchitti

*New York University
Computer Science Department
Courant Institute of Mathematical Sciences*

Agenda

1 RA Vision, Principles, Policies, and Guidelines

2 RA Blueprinting - Business Architecture

3 RA Blueprinting – Application Architecture

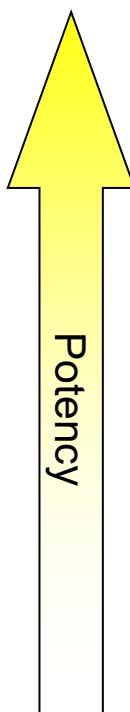
4 RA Blueprinting – Information Architecture

5 RA Blueprinting – Technical Infra. Architecture

6 Appendix A: Business Architecture Details

Vision, Principles, Policies, and Guidelines Definitions

The Reference Architecture Vision, Principles, Policies, and Guidelines help the decision process by setting boundaries. Each of them indicates a different level of emphasis.



- **Vision**
 - Vision is the objective that the organization is striving to achieve. All principles and guidelines enable progress towards the vision.
- **Principles**
 - Principles are the guiding lights to which all policies, guidelines, and decisions should be traceable. From the 'governance' perspective, these are like the constitution, inviolable and relatively stable.
- **Policies**
 - Policies are the rules that establish a minimum set of constraints. These can only be violated with high-level visibility and clearance.
- **Guidelines**
 - Guidelines are like morals that should be followed as a matter of good citizenship; however, there is no severe penalty for violation.

Reference Architecture Vision

Vision

The Reference Architecture is the IT foundation that enables a business to execute its strategic vision

Reference Architecture Vision

The Reference Architecture is the IT foundation that enables a business to execute its strategic vision

Disposition	Vision
Intention	Alignment to Business Strategy
Motivation	<ul style="list-style-type: none">▪ IT is integral to the success of a business vision▪ IT is a business organization that delivers business value using technology▪ Business value is maximized through consistent, effective, and timely delivery of IT solutions▪ Reference Architecture needs to focus on maximizing the strategic business value of IT assets
Implication	<ul style="list-style-type: none">▪ Business needs to articulate and evolve its strategic vision▪ Reference Architecture should evolve along with the strategic business vision▪ Business needs to recognize and include IT as a strategic enabler of its vision▪ Process and governance must be in place to ensure that initiatives align with the vision▪ IT team members need to think of themselves as business people first and technologists second

Vision, Principles, Policies and Guidelines Overview

Vision

- The Reference Architecture is the IT foundation that enables a business to execute its strategic vision

Principles

- IT solutions will align with overall business strategy
- IT individuals are business people first and technologists second
- Architecture Governance approval will be required for all IT solutions
- IT solutions are to be architected with a clear separation of concerns
- IT solutions will be proactively designed for reuse
- The Reference Architecture will be standards based
- All IT assets will have identified business and IT owners
- IT will enable capabilities and services beyond traditional business models
- IT solutions will be driven by a consistent method
- The Reference Architecture will drive evolution of the IT environment

Policies / Guidelines

Application

- Each component, service, process, view, data store and rule will be aligned to a specific business capability and unique business owner
- Interactive business functionality will be exposed through services designed for reuse
- User experience standards will be followed for each medium
- Integration is separated from business functionality
- Reuse existing components instead of implementing new ones
- Production development efforts that do not fall into established guidelines will be coordinated with the architecture office
- Complex business rules and process flow will be abstracted from other application code

Information

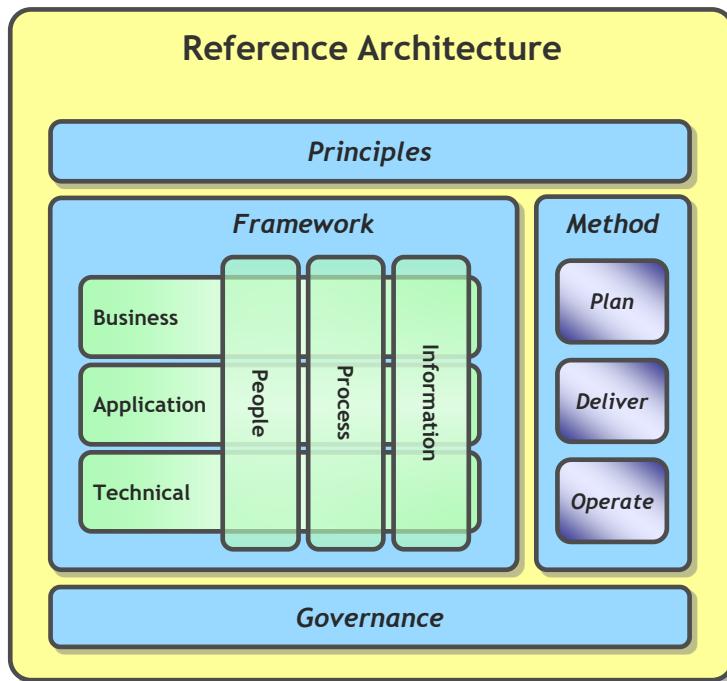
- An enterprise information model will be used to assure a common and logical definition of data
- Data should be treated as a corporate asset
- Data should not be replicated within the same information layer
- Data access and placement should be based on a set of business-wide standards
- Referential Integrity is an attribute of the data layer
- Data stores will meet the service level agreements of each layer of the application architecture
- Data in a Transactional Data Store that is needed in the Data Warehouse is stored first in the Operational Data Store
- The architecture shall support transport of data between and within layers at various levels of granularity and frequency
- Oracle should be used as the database management system for new development

Technology

- Business and application teams will specify the qualities of service expected of the infrastructure instead of the specific infrastructure components
- Infrastructure capabilities should be implemented with commercial off the shelf software (COTS) instead of being developed by the application team
- Infrastructure nodes should be deployed on standard utility platforms and offerings where available
- Infrastructure should implement security capabilities to protect information within and beyond the Enterprise
- Infrastructure services will conform to industry standards where such standards are available
- Implement standardized IT environments that support development, testing, and implementation of applications

Reference Architecture Description

A Reference Architecture consists of foundational principles, an organizing framework, a comprehensive and consistent method, and a set of governing processes and structures.



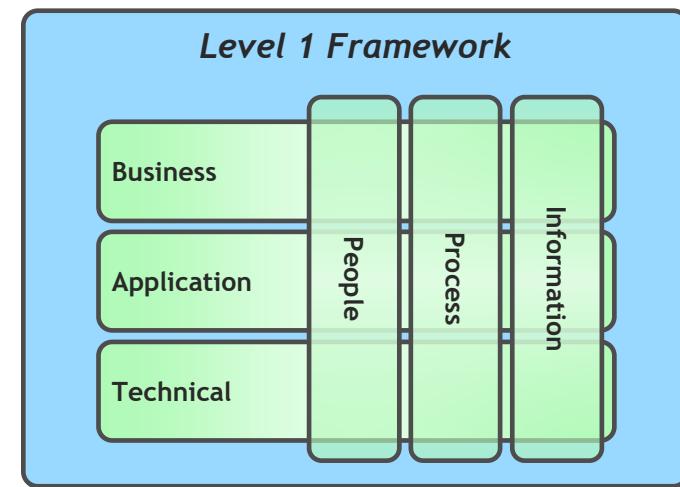
- Principles provide the foundation upon which the Reference Architecture is based. It includes a set of architectural terms as well as numerous principles, policies, and guidelines for governing the architecture.
- Framework is the organizing basis for the Reference Architecture and defines the architectural domains and disciplines that enable separation of concerns and IT to business alignment.
- Method is the comprehensive set of defined repeatable processes that are followed for a consistent and controlled realization of the Reference Architecture.
- Governance is the set processes and organizational structures that ensure conformity to the Reference Architecture.

Reference Architecture Framework (1/2)

The framework is the organizing basis of the Reference Architecture and defines the architectural domains and disciplines that enable architectural separation of concern while also promoting IT to business alignment. The framework is defined at two levels.

Level 1 Framework

- At this higher level (Level 1), the framework defines the three major IT architectural domains of Application, Information, and Technical (also referred to as Infrastructure). These plus the Business, People, and Process domains equate to a traditional EA organizing structure.
- The Business domain is defined by a set of Required Business Capabilities.
- The Application, Information, and Technical Architectures are defined in much greater detail.

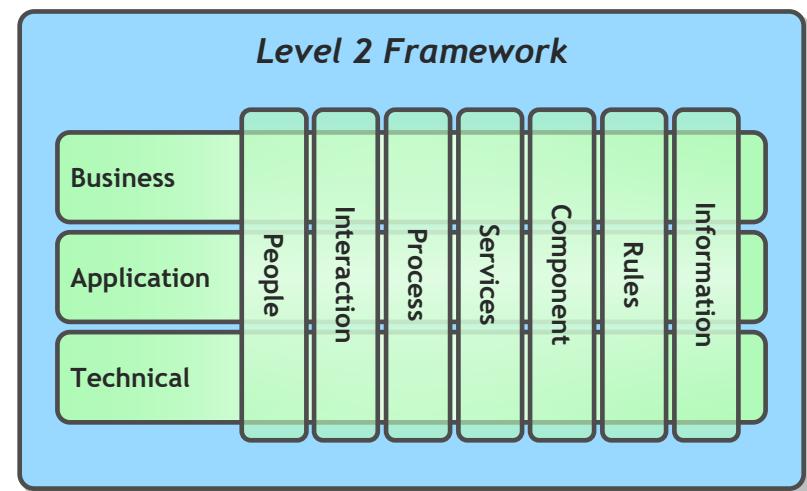


Reference Architecture Framework (2/2)

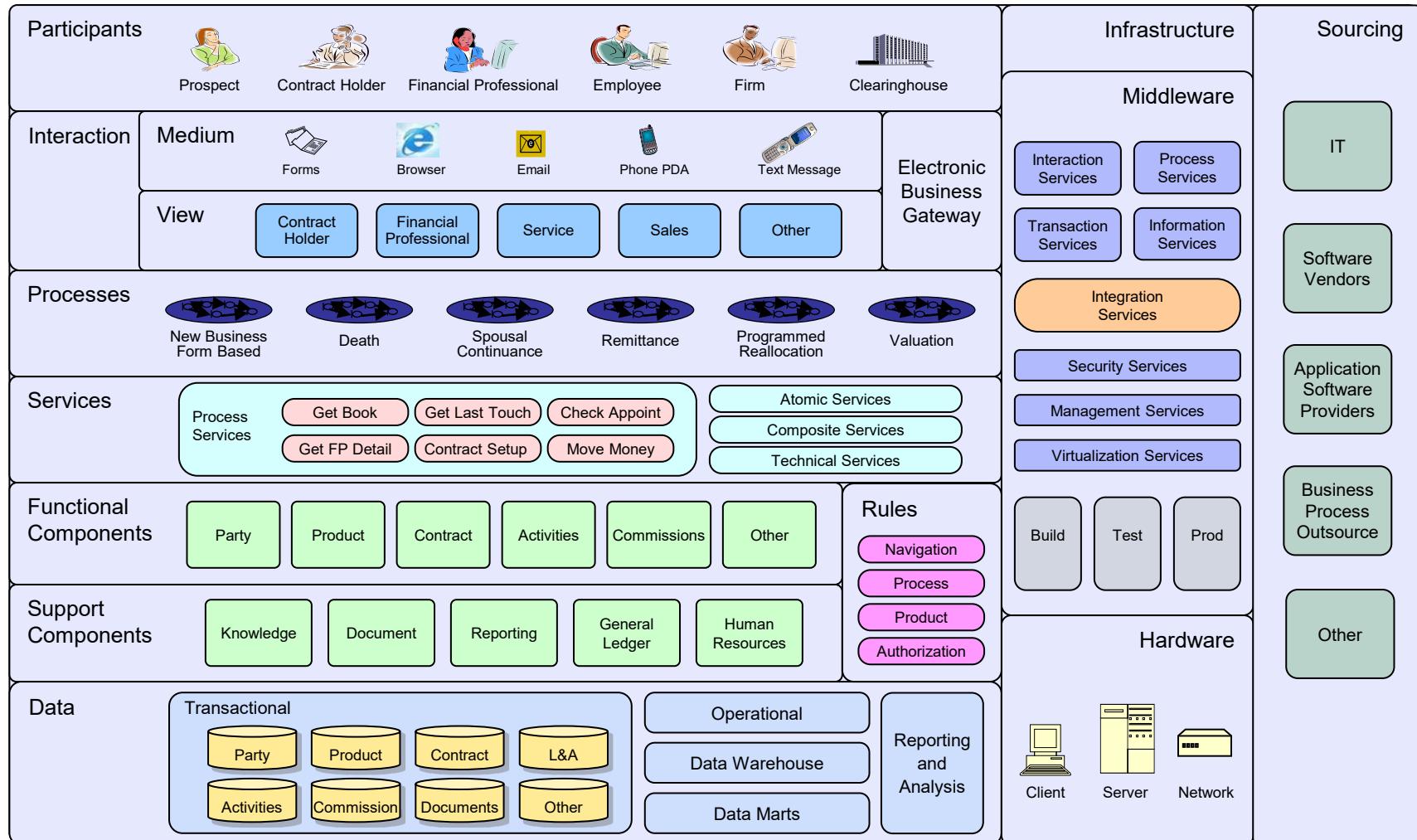
The framework is the organizing basis of the Reference Architecture and defines the architectural domains and disciplines that enable architectural separation of concern while also promoting IT to business alignment. The framework is defined at two levels.

Level 2 Framework

- At this more detailed level (Level 2), the framework further decomposes the People, Process, and Information domains into seven distinct architectural disciplines of People, Interaction, Process, Services, Component, Rules, and Information.
- While all architectural domains (horizontal and vertical) enable and support separation of architectural concerns, it is these seven distinct vertical architectural disciplines that visibly promote alignment between the Business and IT domains.

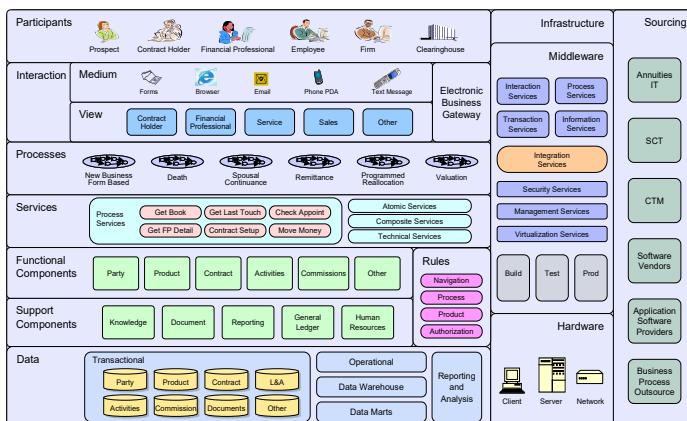


Reference Architecture Blueprint (sample)



Reference Architecture Blueprint Description

The Reference Architecture Blueprint is the single conceptual diagram that illustrates a combined view of the Application, Information, and Technical (Infrastructure) architectures.



- Application Architecture elements are represented as horizontal rows on the left side of the Blueprint and include:
 - Participants consisting of people and organizations
 - Interaction consisting of both mediums and views
 - Processes representing business flow
 - Services representing reusable activities
 - Components identifying groups of like function
 - Rules representing decisions and policies
 - Data representing business information
- The Data row is further detailed to represent the Information Architecture elements including both transactional and analytical dimensions
- Technical Architecture elements are represented on the right side of the Blueprint and include various middleware and hardware capabilities
- Sourcing is also highlighted as a key dimension

Principles

- IT solutions will align with overall business strategy
- IT individuals are business people first and technologists second
- Architecture Governance approval will be required for all IT solutions
- IT solutions are to be architected with a clear separation of concerns
- IT solutions will be proactively designed for reuse
- The Reference Architecture will be standards based
- All IT assets will have identified business and IT owners
- IT solutions will be driven by a consistent method
- The Reference Architecture will drive evolution of the IT environment

IT solutions will align with overall business strategy

Disposition	Principle
Intention	Align all the IT activities and solutions with business strategy as a whole and not just to specific departmental function or immediate needs
Motivation	<ul style="list-style-type: none">▪ Fulfilling departmental needs while aligning with the overall business strategy▪ Aim to derive long term benefits while satisfying immediate needs
Implication	<ul style="list-style-type: none">▪ IT needs to work with business to ensure a documented business strategy▪ Need to guide departmental business needs to align with overall business strategy▪ Reference architecture will be used to align IT solutions with business strategy▪ Need to identify metrics to measure ‘alignment with business strategy’▪ Create a clear funding model to support strategic IT development▪ Minimize tactically built redundant solutions

IT individuals are business people first, technologists second

Disposition	Principle
Intention	Change the culture of the IT organization to act based on business drivers ahead of technology needs to ensure solutions meet business needs effectively and efficiently
Motivation	<ul style="list-style-type: none">▪ Minimizing technology for technology's sake▪ Ensure IT solutions align with business
Implication	<ul style="list-style-type: none">▪ IT team members will have to improve understanding of business▪ Business will need to involve IT in its strategy evolution▪ IT team members are stakeholders in business and not 'order takers'

Architecture governance approval will be required for all IT solutions

Disposition	Principle
Intention	Ensure compliance with reference architecture
Motivation	<ul style="list-style-type: none">▪ Architecture will not be effective if it is not applied consistently▪ Governance will provide valuable feedback for evolution of architecture
Implication	<ul style="list-style-type: none">▪ Architecture approval process will need to be defined▪ Architecture approval process will need to be integrated with the SDLC▪ Solution designers and architects must understand and buy into reference architecture to ensure that governance doesn't get viewed as an overhead

Separation of Concerns Principle

Principles,
Policies, and
Guidelines

IT solutions are to be architected with a clear separation of concerns

Disposition	Principle
Intention	Provide separation of concerns between architecture domains, between architecture layers, and between elements within an architecture layer to ensure that each element delivers only the capability that it is responsible for
Motivation	<ul style="list-style-type: none">▪ Support growth, scalability, resiliency, maintainability, usability, and flexibility in an independent and reusable fashion▪ Ensure components are responsible for functions that they do best▪ Put focus on building reusable ‘capabilities’
Implication	<ul style="list-style-type: none">▪ Architecture needs to be layered▪ Guidelines must be developed to clarify usage of various components▪ Reusable components will have to be defined for various functions

IT solutions will be proactively designed for reuse

Disposition	Principle
Intention	Improve efficiency, timeliness, and reduce the risk of solutions
Motivation	<ul style="list-style-type: none">▪ Ensure that reuse is a strategic driver and not just an afterthought▪ Focus on long term TCO and risk and not just on immediate benefits▪ Enable IT to be viewed as a valued partner to the business by being more efficient and responsive
Implication	<ul style="list-style-type: none">▪ Future requirements will need to be considered while estimating and designing solutions▪ There may be higher initial cost in building reusable solutions▪ A repository of reusable solutions will need to be built▪ Financial model will need to be modified to promote reuse

The Reference Architecture will be standards based

Disposition	Principle
Intention	Improve interoperability, availability of resources, and have an external support ecosystem
Motivation	<ul style="list-style-type: none">▪ Standards permit commoditization resulting in lower costs and greater capabilities▪ Easier integration with external partners▪ Wider availability of resources and components▪ Leverage industry wide expertise▪ Allow IT to focus on business solution delivery
Implication	<ul style="list-style-type: none">▪ Business and technical standards need to be identified, approved, disseminated, and managed▪ Staff needs to be trained in proper usage of standards▪ Governance process will enforce adherence to standards▪ Choosing standards may limit choice of solutions in some areas

All IT assets will have identified business and IT owners

Disposition	Principle
Intention	Ensure point of contacts, one from business and one from IT, exist for decision making on all IT assets
Motivation	<ul style="list-style-type: none">▪ Improve service and control▪ Improve speed of decision making▪ Clear point of escalation for problem resolution, trade-offs, etc.
Implication	<ul style="list-style-type: none">▪ Owners will need to be responsible and accountable for the assets▪ The roles and responsibilities of owners will need to be clearly defined▪ Owners of systems and information will need to be identified▪ Owner decisions will need to align with strategic goals and directions

IT will enable capabilities and services beyond traditional business models

Disposition	Principle
Intention	Support the growth aspiration of the business in terms of acquisition and non-traditional products and distribution
Motivation	<ul style="list-style-type: none">▪ Capture larger share of growing market and enable new markets▪ Effectively and efficiently support growth and innovation from non-traditional models▪ Allow for multi-carrier capabilities across the business and outside the business organization as needed
Implication	<ul style="list-style-type: none">▪ IT will need to effectively and efficiently support third-party administration▪ IT will need to focus on core business offerings and non-traditional products simultaneously IT will need to consider non-traditional implications while delivering traditional offerings

IT solutions will be driven by a consistent method

Disposition	Principle
Intention	Drive IT solutions by a consistent method and not personal preferences
Motivation	<ul style="list-style-type: none">▪ Repeatability of solutions promotes reuse▪ Consistency of methodology will promote efficiency▪ Consistency improves quality
Implication	<ul style="list-style-type: none">▪ A consistent methodology will need to be developed to planning▪ A consistent methodology will need to be developed to delivering▪ A consistent methodology will need to be developed to operating▪ The planning, delivering, and operating methodologies should be synergistic▪ Decision criteria will need to be developed for solution selection

The Reference Architecture will drive evolution of the IT environment

Disposition	Principle
Intention	Drive change through Reference Architecture while coexisting with existing IT assets
Motivation	<ul style="list-style-type: none">▪ Architecture change will be evolutionary not revolutionary▪ Existing assets will be transformed only in response to business needs and at a rate the business can justify
Implication	<ul style="list-style-type: none">▪ A roadmap of evolution will need to be developed▪ Business case will need to be developed for all transformations▪ The Reference Architecture will always be ahead of the current state as it will continue to evolve in response to evolving business and industry needs▪ Benchmarks and Metrics will need to be identified and tracked to measure progress

Agenda

1 RA Vision, Principles, Policies, and Guidelines

2 RA Blueprinting - Business Architecture

3 RA Blueprinting – Application Architecture

4 RA Blueprinting – Information Architecture

5 RA Blueprinting – Technical Infra. Architecture

6 Appendix A: Business Architecture Details



Sample Study – Overall Enterprise Interview Summary

Goals

- Develop innovative products
- Increase gross sales, net sales, and market share
- Improve customer satisfaction while reducing unit cost
- Effectively manage risk while allowing for innovation leadership

Drivers

- New product features are coming to market at ever increasing rate
- Industry and regulatory changes are changing the BU landscape
- Increased touches lead to increased sales
- Channels and partners expect differentiated service levels
- Ease of doing business is directly correlated to sales
- Increasing need to proactively manage risk due to increased rate of product complexity and regulatory oversight

Sample Study – Overall Enterprise Interview Summary

Opportunities

- Improve data and analytics capabilities in all functional areas
- Increase speed and flexibility to enable rapid innovation of products and features, but not at cost of quality
- Improve ease of doing business by tailoring product, sales, and service to client specific needs
- Align business objectives across product, sales and service
- Develop a foundation and framework to guide and govern the BU IT initiatives
- Effectively invest in IT to catch up with industry and competition
- Improve IT capabilities by better aligning technologies, resources, and functions
- Enhance solution development lifecycle (SDLC)
- Influence standards and proactively build skills through early involvement in industry efforts

Sample Study – Business Capability Areas



Sample Study – Product Area Interview Summary (1/2)

Goals

- Develop innovative products and features that are first to market or fast followers
- Develop new products to capture non-traditional markets and distribution channels
- Balance innovation and competitiveness with risk exposure
- Metrics are gross and net sales, market share, and profitability

Drivers

- Innovation is focused on living benefits
- Product chassis is fairly stable for traditional products
- First mover advantage is continuously shrinking
- New product features are coming to market at ever increasing rate
- Financial planners are looking for innovative products to transfer assets

Sample Study – Product Area Interview Summary (2/2)

Opportunities

- Develop a holistic, integrated end-to-end product life-cycle capability
- Increase system configurability to enable rapid innovation of products and features
- Increased ability for analytic techniques and modeling
- Increase resource efficiency by better aligning resources to value added tasks
- Better facilitate asset retention through innovative product features
- Develop a product platform that supports differentiation by firm and channel

Sample Study – Sales Area Interview Summary (1/1)

Goals

- Increase gross sales, net sales, and market share
- Increase market share in warehouse channel while continuing to grow other channels
- Increase depth and penetration within each distribution partner

Drivers

- BU instruments are sold not bought
- Increased touches lead to increased sales
- High percentage of assets under management turnover annually
- Acquisitions may enhance and/or disrupt existing distribution models

Sample Study – Sales Area Interview Summary (2/2)

Opportunities

- Develop a holistic view of clients, territories, and channels
- Deliver the right information to the right stakeholder at the right time
- Improve sales process efficiencies
- Better manage asset turnover by focusing on retention

Sample Study – Service Area Interview Summary (1/1)

Goals

- Improve customer satisfaction by being more flexible and easier to do business with
- Reduce unit cost of service
- Offer consistently good service

Drivers

- Channels and partners expect differentiated service levels
- Ease of doing business is directly correlated to sales
- Industry is standardizing straight through processing
- Key industry vendors and firms have significant impact on servicing and processing

Sample Study – Service Area Interview Summary (2/2)

Opportunities

- Increased focus on service could greatly enhance customer satisfaction and efficiency
- Satisfy different service level expectations for different channels
- Leverage market presence to influence industry direction
- Redefine business processes and automation to better match current market position, scale, and strategies
- Integrate service data with other parts of organizations
- Dashboard visibility into the current service process

Sample Study – Client Focus Area Interview Summary (1/2)

Goals

- Improve customer satisfaction by being more flexible and easier to do business with
- Increase assets under management through product innovation and improved asset retention
- Raise market share and penetration with FP's through improved analytics and activity management
- Provide real-time client information to wholesalers to enable deeper relationships

Drivers

- Increased touches and deeper knowledge lead to increased sales
- Ease of doing business is directly correlated to sales
- 70% of annuity investments are TOA
- The focus is on penetration with FP's as most major firms are already onboarded

Sample Study – Client Focus Area Interview Summary (2/2)

Opportunities

- Integrating customer experience information from various sources to create a holistic understanding of client
- Increased focus on client-centric service could greatly enhance customer satisfaction and efficiency
- Understanding leading indicators of asset turnover to take timely corrective actions
- Satisfy different service level expectations for different channels
- Better align product features to enhance asset retention

Sample Study – Risk and Financial Management Interview Summary (1/1)

Goals

- Effectively manage risk while allowing for innovation leadership
- Assure the overall operational and financial health of the organization through accurate measurement and forecasting
- Ensure compliance to all external regulations and internal policies
- Safeguard confidential information about our clients and company

Drivers

- Regulatory changes will continue to shape the BU landscape
- Increasing need to proactively manage risk due to increased rate of product complexity and regulatory oversight
- Current monitoring techniques can no longer support scale of business operations
- Rate of innovation drives the need for faster financial forecasting

Sample Study – Risk and Financial Management Interview Summary (2/2)

Opportunities

- Invest in comprehensive business continuity and information security
- Enable business relationships and innovation through relaxed regulatory and legal interpretations while maintaining acceptable risk
- Improve efficiency, accuracy and flexibility of financial forecasting
- Proactively help external business partners manage their risk
- Improve platform flexibility and extensibility to support innovation and complexity of business
- Migrate to rules-based and extensible systems to support complexity introduced by Innovation

Sample Study – Information Technology Area Interview Summary (1/1)

Goals

- Match IT capability and capacity to fulfill business needs at the speed and schedule they evolve
- Increase efficiency of IT solution delivery through standardized environment and tools
- Create common purpose/mission for organization through a well crafted, fully baked, collaborative architecture strategy.

Drivers

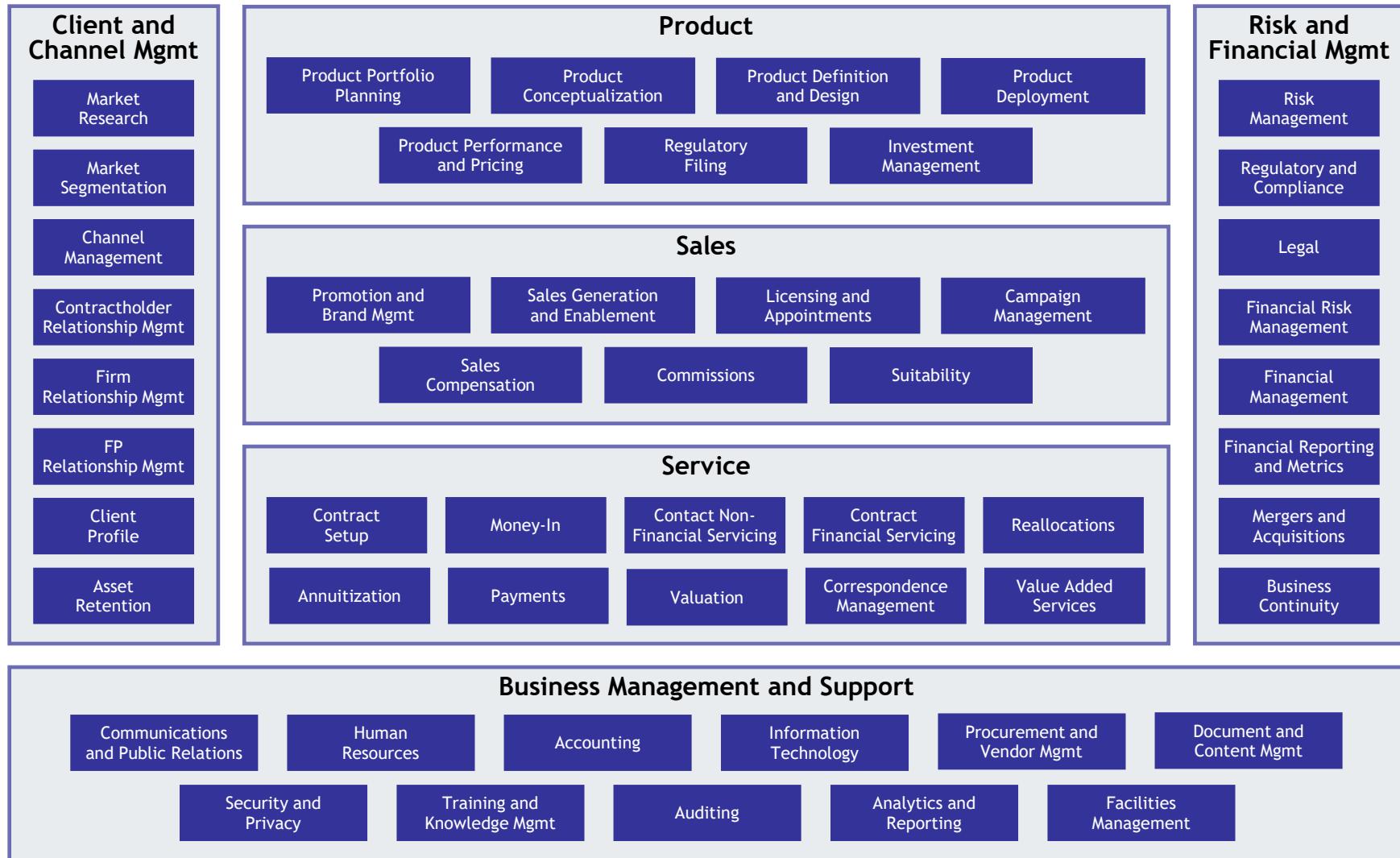
- Business is looking for immediate and continuous improvements in IT value statement
- Continuous business focus on tactical IT solutions to fulfill immediate needs with minimum short term investment
- Acquisition and unique opportunities are key strategies for exponential growth
- Legacy systems and contractual considerations will influence organizational decisions
- IT is increasingly essential to remain competitive
- Partners will continue to play an increasing role and force technology decisions including standards

Sample Study – Information Technology Area Interview Summary (2/2)

Opportunities

- Increase resource efficiency by better aligning resources to value added tasks
- Improve solution development lifecycle (SDLC) and alignment to business timelines
- Maximize the effective use of tools and technologies
- Satisfy increased demand for data, analytic tools, and computational capacity
- Reinvest efficiency savings into IT to accelerate rate of improvement
- Influence standards and proactively build skills through early involvement in industry efforts
- Motivate individuals to adhere to strategic processes and technologies defined in reference architecture

Resulting Business Capabilities



Product Area

Required Capabilities

- ❑ Product Portfolio Planning
 - Product Market Alignment
 - Feature Analysis
 - Product Opportunities
 - Product Release Strategy
- ❑ Product Conceptualization
- ❑ Product Definition and Design
 - Product Requirement Analysis
 - Product Design
 - Product Scoping
- ❑ Product Deployment
 - Product Setup
 - Product Training
 - Marketing Material Development
 - Product Launch
- ❑ Product Performance and Pricing
 - Actuarial
 - Product Pricing
 - Product Performance Analysis
 - Product Profitability Analysis
- ❑ Regulatory Filing
 - Filing Preparation
 - Filing
- ❑ Investment Management
 - Investment Relationship Management
 - New Investment Management Strategy
 - Investment Risk Analysis
 - Asset Allocation Selection
 - Fund Manager Selection
 - Alternate Investment Strategy

Sales Area

Required Capabilities

- ❑ Promotion and Brand Management
 - Brand and Penetration Analysis
 - Brand Awareness
 - Brand Promotion
- ❑ Sales Generation and Enablement
 - Sales Generation Ideas
 - Sales Education
 - Sales Lead Generation
 - Illustrations
 - Sales Materials
- ❑ Licensing and Appointments
 - Licensing
 - Appointment
- ❑ Campaign Management
 - Campaign Planning
 - Campaign Management
 - Campaign Execution
- ❑ Sales Compensation
 - Wholesaler Comp Strategy and Planning
 - Wholesaler Comp Setup
 - Wholesaler Comp Calculations
- ❑ Commissions
 - FP Comp Strategy and Planning
 - FP Comp Setup
 - FP Comp Calculations
 - FP Comp Preference Setup
- ❑ Suitability
 - Suitability Review of Transactions

Service Area (1/2)

Required Capabilities

- ❑ Contract Setup
 - Order Entry
 - Exception (NIGO) Handling
 - Contract Issuance
- ❑ Money-In
 - New Business Collections
 - Commission Recapture
 - Sub-pay Collections
- ❑ Contract Non-financial Servicing
 - Contract Administration
 - Transfer of Block
 - Third Party Administration
- ❑ Contract Financial Servicing
 - Valuation
 - Contract Surrender
 - Payout Processing
 - Outbound Exchange Processing
 - Death Benefits
 - 72T/ 72U Exchanges
 - Partial Withdrawals
 - Cancellation
- ❑ Reallocations
 - Program Money Movement
 - Client Directed Money Movement
- ❑ Annuitization
 - Contract Maturity
 - Immediate Annuitization
- ❑ Payments
 - Moneys Out
 - Annuity Payments
 - Exchange Payments
 - Compensation Distribution

Service Area (2/2)

Required Capabilities

- ❑ Payments
 - Moneys Out
 - Annuity Payments
 - Exchange Payments
 - Compensation Distribution
- ❑ Valuation
 - Valuation
- ❑ Correspondence Management
 - Generate Correspondence
 - Assemble Correspondence
 - Issue Correspondence
- ❑ Value Added Services
 - Non-Contract Service Planning
 - Non-Contract Service Design
 - Non-Contract Service Execution
 - TPIA Servicing

Client and Channel Management (1/2)

Required Capabilities

- ❑ Market Research
 - Industry Analysis
 - Client Analysis
 - Competitive Analysis
 - Existing and Trends
 - Client Behavior Modeling
 - Market Maturity/ Readiness Analysis
- ❑ Market Segmentation
 - Contractholder Segmentation
 - Firm Segmentation
 - FP Segmentation
- ❑ Channel Management
 - Channel Strategy
 - Channel Segmentation and Planning
 - Acquisition of New Firms
 - Firm Onboarding
- ❑ Contractholder Relationship Management
 - Contractholder Relationship Strategy
 - Contractholder Relationship Management
 - Contractholder Satisfaction Management
 - Contractholder Activity Management
- ❑ Firm Relationship Management
 - Firm Relationship Management
 - Firm Activity Management
 - Firm Satisfaction Management
 - Firm Effectiveness, Performance & Profitability
 - Product Release Planning by Firm

Client and Channel Management (2/2)

Required Capabilities

- ❑ FP Relationship Management
 - FP Relationship Management
 - FP Activity Management
- ❑ Client Profile
 - Contractholder Profile Setup
 - Contractholder Profile Maintenance
- ❑ Asset Retention
 - Asset Retention Strategy & Planning
- FP Satisfaction Management
- FP Effectiveness, Performance & Profitability
- Firm Profile Setup and Maintenance
- FP Profile Setup and Maintenance
- Execution

Risk and Financial Management (1/2)

Required Capabilities

- ❑ Risk Management
 - Environmental Risk Management
 - Operational Risk Management
 - Overall Risk Management
- ❑ Regulatory and Compliance
 - Regulatory Monitoring
 - Product Regulatory Analysis
 - Regulatory Compliance (AML, OFAC)
 - Proactive Regulatory Sensing
- ❑ Legal
 - Legal Interpretation
 - Legal Environment Monitoring
 - Litigation Management
- ❑ Financial Risk Management
 - Market Risk Management
 - Actuarial Financial Risk Analysis
 - Reserving
- ❑ Financial Management
 - Financial Policy
 - Capital Investment Management
 - Fund Trades
 - Treasury
 - Forecasting

Risk and Financial Management (2/2)

Required Capabilities

- ❑ Financial Reporting and Metrics
 - Statutory Reporting
 - Internal Financial Reporting
 - External Non-Regulatory Reporting
- ❑ Merger and Acquisition
 - M&A Strategy and Planning
 - M&A Opportunity Identification
 - M&A Bids and Proposals
 - M&A Execution
 - Acquire and Transfer of Block of Business
 - Carrier Acquisition
- ❑ Business Continuity
 - Business Continuity Planning
 - Business Continuity Coordination
 - Disaster Recovery

Business Management and Support (1/2)

Required capabilities

- Communications and Public Relations
 - Communications Planning
 - Communications Management
 - Perception Analysis
 - Crisis Management
 - Internal Communications
- Security and Privacy
 - Physical Security Management
 - Information Security Management
 - Intellectual Property Management
 - Privacy Management
- Human Resources
 - Staffing
 - Personnel Retention
 - Personnel Compensation
 - Employee Relations
 - Employee Benefits
- Training and Knowledge Management
 - Knowledge Need Analysis
 - Training Development
 - Training
 - Knowledge Harvesting
 - Channel Education Support
- Accounting
 - GL
 - AP
 - AR
 - Budgeting
 - Tax
 - Cash and Payment Reconciliation

Business Management and Support (2/2)

Required capabilities

- ❑ Auditing
 - Financial Auditing
 - Operational Auditing
 - Security and Privacy Auditing
 - SOX Financial Auditing
- ❑ Information Technology
 - Plan Information Systems
 - Build Information Systems
 - Operate Information Systems
 - Assure Information Systems
- ❑ Analytics and Reporting
 - Operational Reporting
 - Project Portfolio Reporting
 - Dashboarding
 - Performance Reporting
- ❑ Procurement and Vendor Management
 - Vendor Relationship Management
 - Procurement
- ❑ Document Management
 - Content Mangement
 - Document Management
 - Image Services
 - Print Services
- ❑ Facilities Management
 - Facilities Acquisition
 - Facilities Maintenance

Information Technology

Required Capabilities

- ❑ Plan
 - Project Portfolio Planning
 - IT Budgeting and Forecasting
 - Reference Architecture
 - Program Management
 - Project Management
- ❑ Build
 - Solution Architecture
 - Solution Design
 - Solution Construction
 - Solution Deploy
- ❑ Operate
 - Problem Management
 - Infrastructure Support
 - Production Support
 - Information Management
- ❑ Assure
 - Software Quality Assurance
 - Metrics
 - Financial Monitoring
 - Governance

Agenda

1 RA Vision, Principles, Policies, and Guidelines

2 RA Blueprinting - Business Architecture

3 RA Blueprinting – Application Architecture

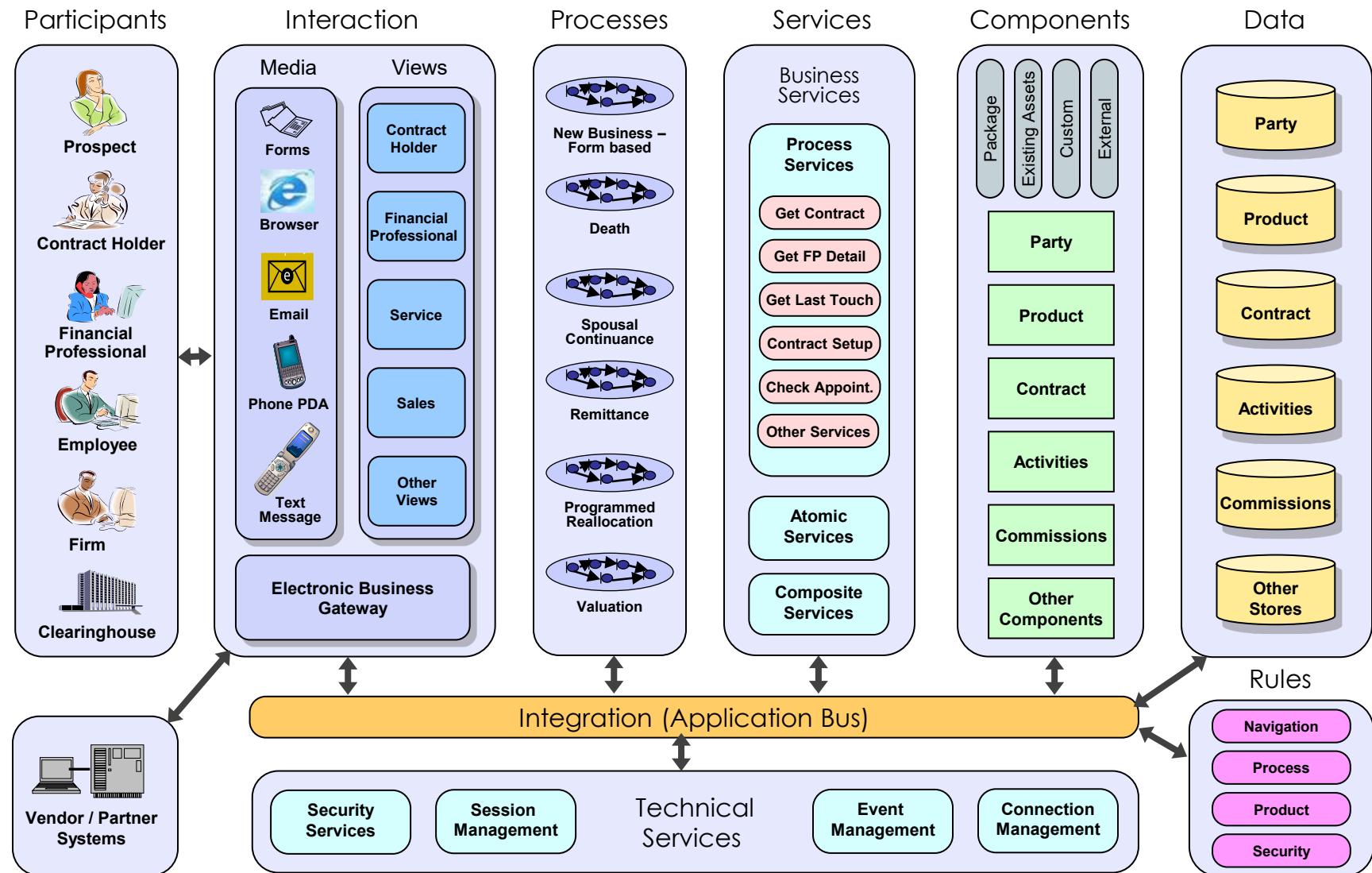
4 RA Blueprinting – Information Architecture

5 RA Blueprinting – Technical Infra. Architecture

6 Appendix A: Business Architecture Details

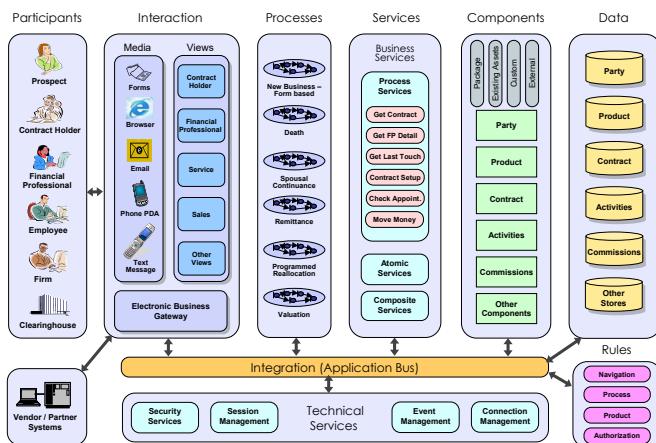


Application Architecture Diagram



Application Architecture Description

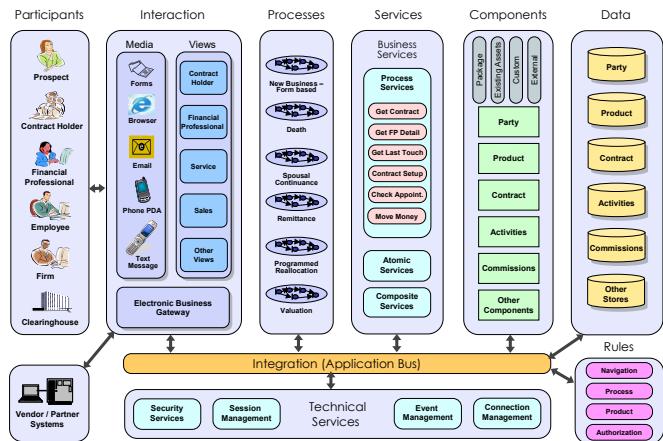
The application architecture describes the logical structure of software supporting and aligned to business operations. The application architecture runs on top of the infrastructure architecture and depends on the information architecture for persistent business data.



- Participants are human or system users of business functionality
- Media are the tools used by users to interact with business systems
- Views present business functionality for specific users
- Processes coordinate sequences of business functions
- Services define simple or complex business functions designed for reuse
- Components are collections of business functionality in a single business area
- Domains are software implementations of business functionality
- Data stores maintain persistent business data
- Rules describe externalized business decisioning criteria
- Integration manages the routing of a request for reusable functionality
- Technical services provide support to building business components

Application Architecture Relationships

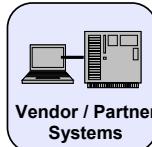
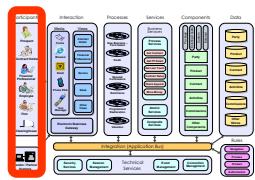
The layers of the architecture interact with each other to support business operations.



- Participants manipulate views through media
- The views manage interactive flow and delegate business functionality to components through services
- In order to invoke business functionality, the view passes a service request to the integration layer
- The integration layer determines which component exposes the service and passes the request to that component
- The component determines internally how to process the business request, processes the request and creates a response
- As a part of processing a request, a component may use persistent business data in an associated data store
- If the business request requires multiple steps of business functionality, the component can use processes to describe the flow between steps

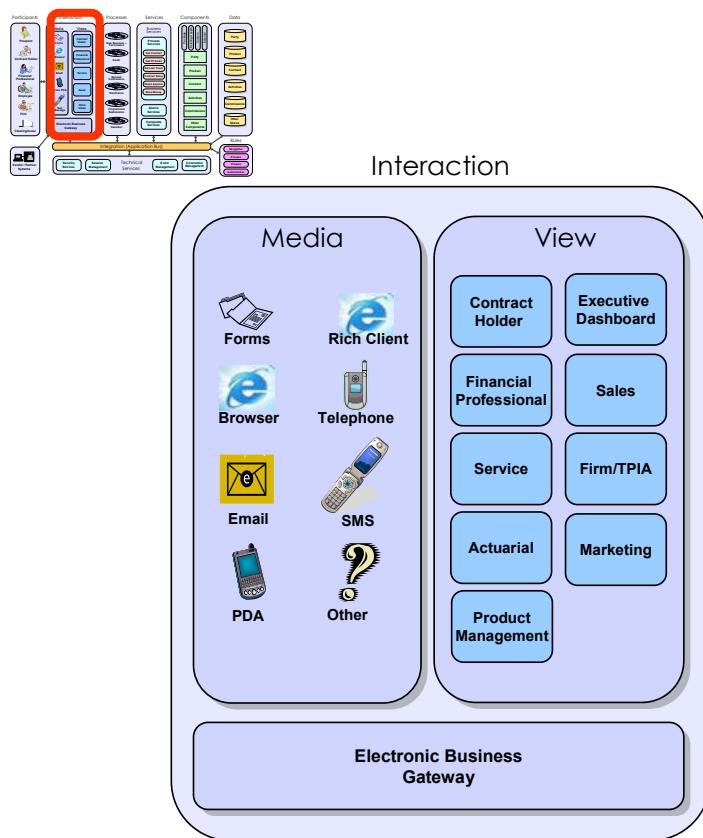
Participants

Participants are human or system users of business functionality.



- Participants can be internal or external to the company
- People participants can include Financial Professionals, Contract Holders, employees, beneficiaries, and others
- Company participants include IBDs, wirehouses, clearinghouses, TPIAs, banks and others
- External system participants use Corporate data or system functions on behalf of other participants

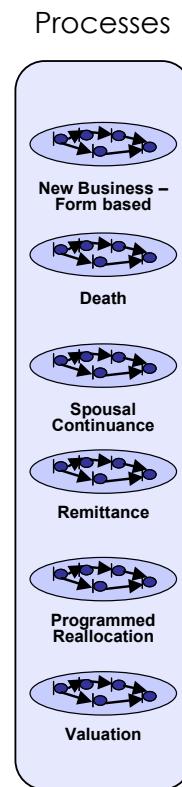
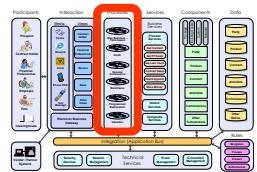
Interaction describes the mechanisms participants use to access business functionality.



- Medium describe the physical devices and standards used for user interactions
- Views are a collection of business functionality and dedicated user interface elements for specific business roles
- The Electronic Business Gateway is the entry point for external consumers of business services

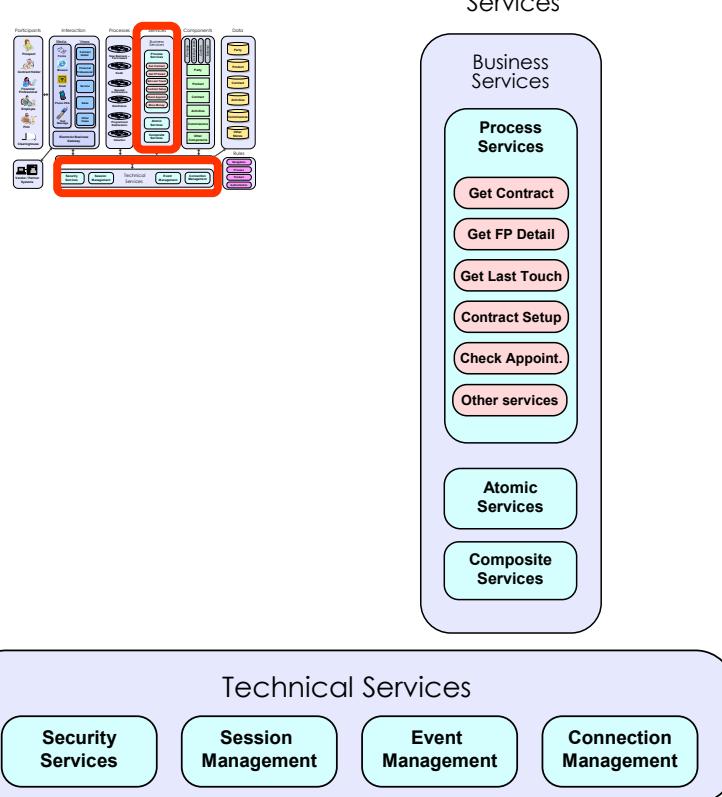
Processes

Processes coordinate sequences of business functions.



- Processes use services to perform individual steps
- Processes may include decision points
- Processes in turn may be used by or exposed as services
- Process may be short- or long-lived
- Processes are intended to be rapidly reconfigurable to reflect changes in the business flow

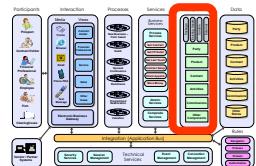
Services define simple or complex business functions designed for reuse.



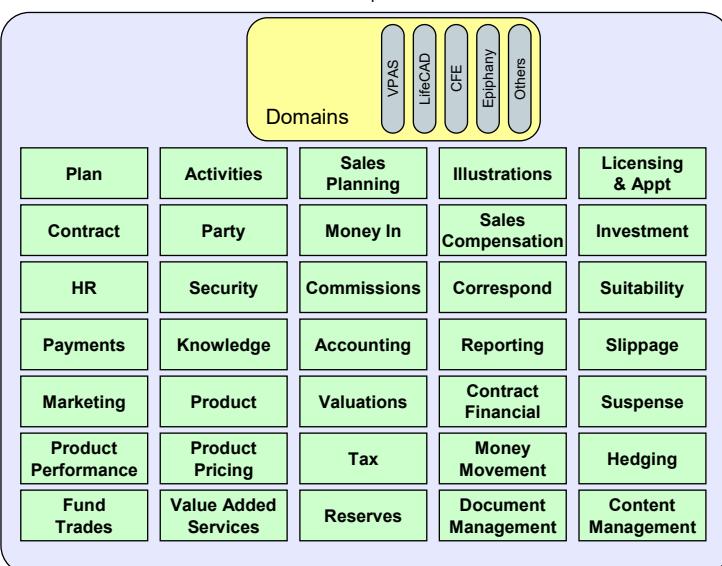
- Each service is designed for reusability and is generally standards based
- Industry standards are used for the data and message model
- Technical standards are used for protocols and packet formats
- **Atomic Services** expose a reusable business activity within a single component
- **Composite Services** expose a business process that leverages multiple services, perhaps across components
- **Technical Services** expose generic support functions that are reusable but independent of specific business alignment, such as logging, authorization, etc.

Components and Domains

Components are collections of business functionality in a single business area.

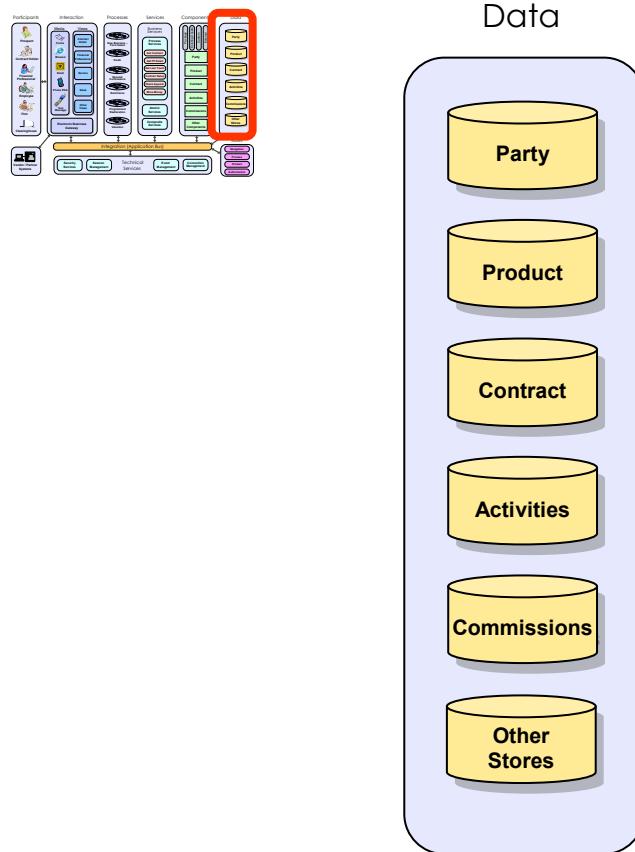


Components



- Components are aligned with one and only one business capability
- Components are described by one or more services
- Components may leverage multiple domains including existing assets, package software, and new development
- A Domain is a collection of business functionality on a single technology platform that may partially or completely implement one or more components

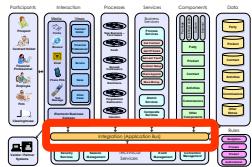
Data stores maintain persistent business data.



- Data includes the transactional, operational, structured, unstructured, and analytical data
- The Data Layer is described in the Information Architecture

Integration

Integration manages the routing of a request for reusable functionality.

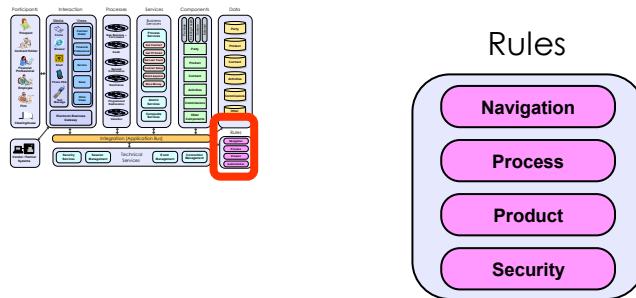


Integration (Application Bus)



- The integration provides an isolation that prevents a service consumer from having to know the exact location of a service provider
- Routing provides a connection between service providers and service consumers
- Registry provides a mechanism for advertising services
- Transform allows for protocol or data packet format changes
- Correlation allows responses to be composed out of multiple service calls
- Monitoring logs traffic patterns
- Point-to-point connections are discouraged

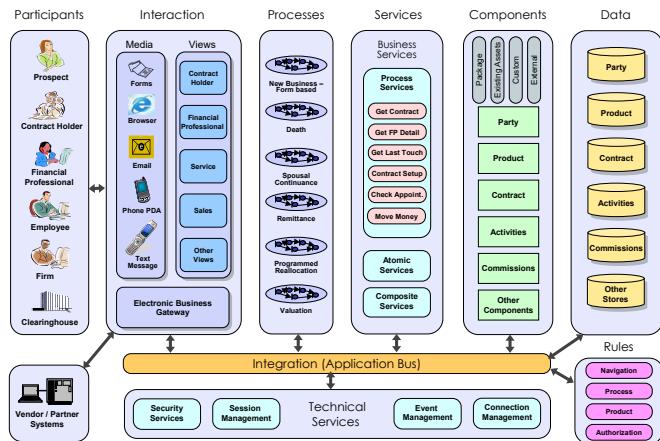
Rules describe externalized business decisioning criteria.



- Rules give the business flexibility by externalizing parameters or business logic
- Rules may be within the same technology platform as other code, or may be in a specialized rules product

Application Architecture Policies

Policies are general rules that must be followed when developing solutions.



- Exceptions must be approved by the Architecture Governance board
- Policies lead to patterns and guidelines which are specific development recommendations

Policies / Guidelines

Application

- Each component, service, process, view, data store and rule will be aligned to a specific business capability and unique business owner
- Interactive business functionality will be exposed through services designed for reuse
- User experience standards will be followed for each medium
- Integration is separated from business functionality
- Reuse existing components instead of implementing new ones
- Production development efforts that do not fall into established guidelines will be coordinated with the architecture office
- Complex business rules and process flow will be abstracted from other application code

Each component, service, process, view, data store and rule will be aligned to a specific business capability and unique business owner

Disposition	Policy
Intention	Ensure that business is aligned with and committed to IT solution
Motivation	<ul style="list-style-type: none">▪ Solutions need business owners that have a vested interest in ensuring functionality is properly described and implemented
Implication	<ul style="list-style-type: none">▪ Components cannot have functionality that cross multiple business capabilities▪ There will be a large number of components with very focused sets of business functionality

Interactive business functionality will be exposed through services designed for reuse

Disposition	Policy
Intention	Ensure business functionality can be used by any consumer regardless of presentation or business alignment
Motivation	<ul style="list-style-type: none">▪ Reduce overall portfolio complexity▪ Provide consistent business functionality implementations across multiple uses of the functionality
Implication	<ul style="list-style-type: none">▪ Business logic is isolated from presentation logic▪ Service definition has to be done with an eye toward alternative uses▪ Batch logic may not be available as a service

User experience standards will be followed for each medium

Disposition	Policy
Intention	Ease user interactions with the system through consistency
Motivation	<ul style="list-style-type: none">▪ Reduce training costs▪ Reduce errors due to inconsistent interfaces
Implication	<ul style="list-style-type: none">▪ Standards have to be defined for each presentation technology▪ User interface design reviews will have to be performed▪ Different media will have different user interface standards

Integration is separated from business functionality

Disposition	Policy
Intention	Business functionality should not know the specific location or implementation of external functionality they depend upon
Motivation	<ul style="list-style-type: none">▪ Facilitate the evolution of components to different platforms
Implication	<ul style="list-style-type: none">▪ Components can be used only through defined service interfaces▪ Consistent data representations have to be used

Reuse existing components instead of implementing new ones

Disposition	Policy
Intention	Consolidate functionality into a single reusable component
Motivation	<ul style="list-style-type: none">▪ Reduce the complexity of the overall portfolio▪ Enhance consistency of business operations
Implication	<ul style="list-style-type: none">▪ Published service definitions have to be reviewed when designing components▪ Users of functionality have to be tracked to address regression issues▪ Two services with similar business functionality should be considered for consolidation▪ Support services (such as authorization and logging) should be used in preference to building equivalent functionality

Production development efforts that do not fall into established guidelines will be coordinated with the architecture office

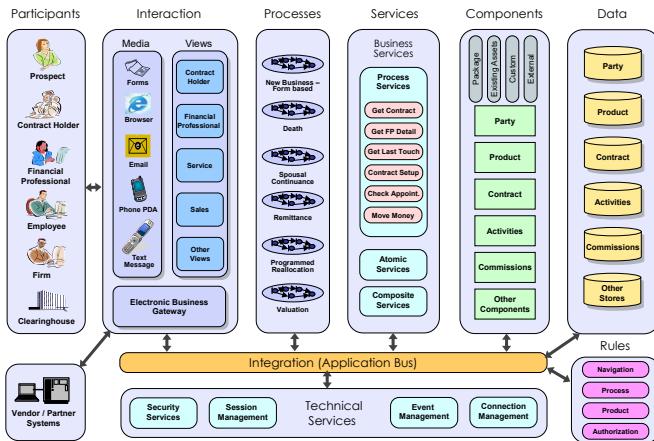
Disposition	Policy
Intention	Allow architecture evolution in a controlled manner
Motivation	<ul style="list-style-type: none">▪ The application architecture must remain consistent to be effective but also has to change to reflect new technologies
Implication	<ul style="list-style-type: none">▪ Project aligned architects and designers do not have the ability to ignore guidelines▪ Experimentation for learning purposes is still allowed and encouraged▪ The architecture office must remain open to new alternatives

Complex business rules and process flow will be abstracted from other application code

Disposition	Policy
Intention	Allow specialized job roles to maintain the most complex parts of the portfolio
Motivation	<ul style="list-style-type: none">▪ Manage the complexity of software implementations by externalizing the most complex portions of the code▪ Enable non-traditional development of rules and process flows
Implication	<ul style="list-style-type: none">▪ Complex business rules and processes may be managed by business analysts rather than developers▪ Rules and process engines need to be evaluated as emerging technology

Application Architecture Patterns

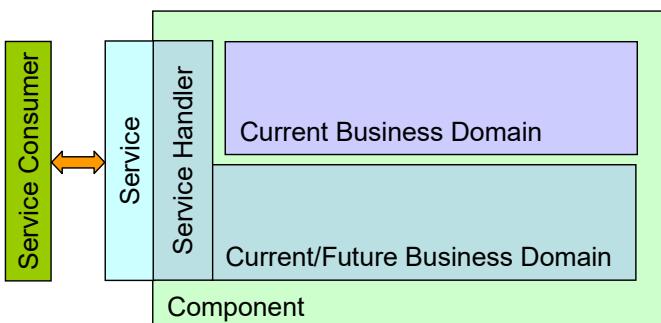
Application Architecture Patterns describe the preferred or accommodated ways of implementing or using architectural constructs.



- All patterns are preferred unless otherwise described
- Developers must consult with the architecture office before developing code that does not follow preferred or accommodated patterns
- The most applicable pattern should be used in preference to “stretching” the intent of a pattern in order to fit to a particular coding style
- Developers should consult with the architecture office if the pattern being used is not a clear match

The component/service pattern is used to expose business functionality for interactive use.

- This pattern allows a component to provide business functionality to a service consumer through a service
- The service handler internal to the component has the responsibility for accepting the incoming service request, parsing it, and determining the appropriate component code to call
- Component functionality may be in the same domain as the service handler or may be an external domain through an adapter
- The service consumer can only use business functionality of the component through services
- All the services exposed by a component share a common data model and business capability



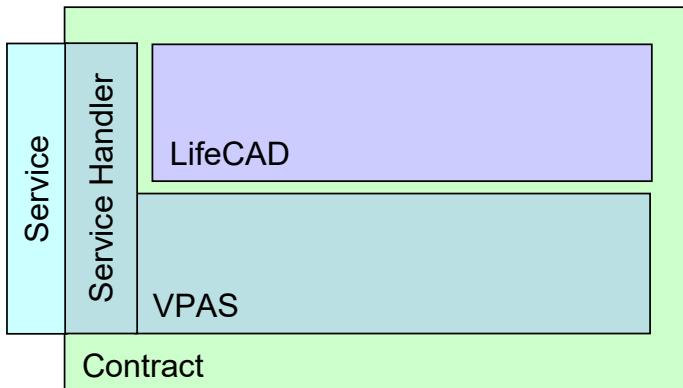
Component/Service is the preferred pattern for interactive use

Disposition	Guideline
Intention	Expose all interactive business functionality as services
Motivation	<ul style="list-style-type: none">▪ Consistent business functionality implementation▪ Hiding of the specific technology platform for a business function
Implication	<ul style="list-style-type: none">▪ Existing domains will have to be wrapped by component and services▪ Users of business functionality will have to invoke the functionality through services.▪ Interactive use can be user or system driven▪ See also<ul style="list-style-type: none">▪Batch pattern▪Component interaction pattern▪External exposure pattern

Component/Service Pattern Example

Example

This example demonstrates how the existing domains of VPAS and LifeCAD will be exposed through components and services.



- VPAS and LifeCAD are large multifunction domains that have business functionality across multiple business capabilities
- Each component (Contract in this case) will expose a subset of that functionality
- The component will expose and rationalize functionality from both systems
- In this example the service handler is in the VPAS domain. The LifeCAD domain is separate
- To access the VPAS contract functionality, a façade is used
- To access the LifeCAD contract functionality, an adapter is used

Progress and Java/J2EE are the preferred component development environments

Disposition	Guideline
Intention	State the preferred component tooling
Motivation	<ul style="list-style-type: none">▪ Standardize development
Implication	<ul style="list-style-type: none">▪ All other development environments are accommodated or discouraged▪ Other existing development environments are accommodated▪ Other development environment which do not have existing assets, or for development outside the existing asset, are discouraged and should be coordinated with the Architecture Office

Strategic business logic development in a component is limited to a single environment

Disposition	Guideline
Intention	Avoid complexity
Motivation	<ul style="list-style-type: none">▪ Developers need guidance to select between the two preferred development environments
Implication	<ul style="list-style-type: none">▪ Each component will have Progress or Java/J2EE as the preferred development environment.▪ All other development environments will be accommodated or discouraged for that component

Application logic and data storage and manipulation are logically distinct from each other

Disposition	Guideline
Intention	Ensure proper isolation
Motivation	<ul style="list-style-type: none">▪ Standardize development
Implication	<ul style="list-style-type: none">▪ Stored procedures should not be used to implement business logic

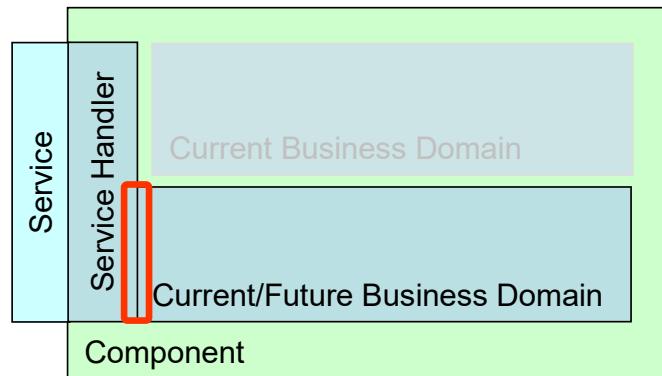
Development environment features not explicitly covered by architecture patterns are discouraged

Disposition	Guideline
Intention	Enforce architectural alignment
Motivation	<ul style="list-style-type: none">▪ Development environments may have features that are not aligned with architectural guidelines▪ Architecture compliance is critical to strategic goals
Implication	<ul style="list-style-type: none">▪ All development must use the same patterns across development environments▪ Development patterns that do not follow architecture patterns are discouraged▪ Non-production learning of development features is approved▪ Architecture patterns have to be well understood

Services are used to expose reusable functionality

Disposition	Guideline
Intention	Align services with reusable business or technical capabilities
Motivation	<ul style="list-style-type: none">▪ Services should expose and protect business functionality▪ Technical services are used to expose widely reusable business enabling functions▪ Discourage the exposure of low-level functions, such as direct data access, as services
Implication	<ul style="list-style-type: none">▪ Stored procedures should not be directly exposed as services even when the technology platform supports this functionality▪ See also<ul style="list-style-type: none">▪ Database Purity Guideline

The domain façade pattern is used within a component where the service handler is implemented in the same domain as the business functionality.

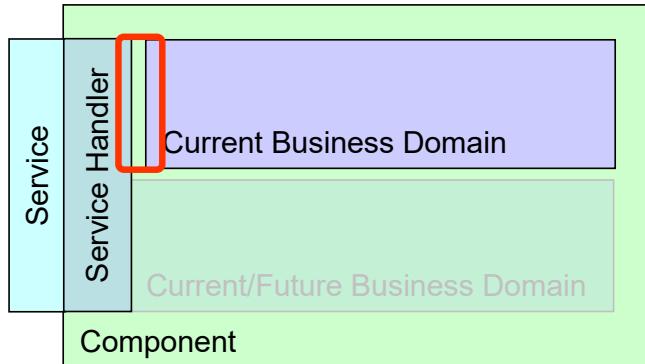


- A façade is lightweight piece of code that hides the details about the exact location and calling conventions of the called module
- Normal platform calling conventions are used
- The data model used for the call may be internal to the domain
- The service handler has the responsibility of ensuring the data model it exposes is consistent with other exposed business functionality
- A façade is used between the service handler and the business functionality to isolate the service handler from the business functionality

Use a façade to isolate the service handler from business functionality in the same domain

Disposition	Guideline
Intention	Enable the business functionality to be moved to a different domain with minimal rework of the service handler.
Motivation	<ul style="list-style-type: none">▪ Reducing maintenance costs and time to market▪ Allow code to be moved to another domain with minimal changes to the service handler
Implication	<ul style="list-style-type: none">▪ Service handler code cannot process business functionality directly▪ See also<ul style="list-style-type: none">▪ Domain Adapter Guideline

The domain adapter pattern is used within a component where the service handler is implemented on a different domain as the business functionality.

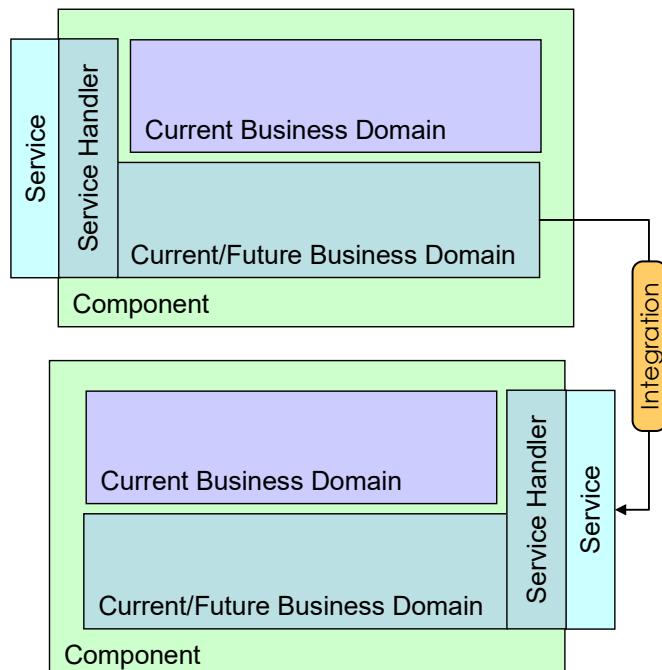


- An adapter is a piece of code that provides an alternative interface to a called module without adding additional business logic
- The most efficient shared communication mechanism should be used
- The data model may be the data model exposed by the service handler or the data model exposed by the existing business domain
- The adapter has the responsibility of ensuring the data model it exposes to the service handler is consistent with other exposed business functionality

Use an adapter to isolate the service handler from business functionality across domains.

Disposition	Guideline
Intention	Enable the business functionality to be moved with minimal rework of the service handler
Motivation	<ul style="list-style-type: none">▪ Reducing maintenance costs and time to market▪ Simplifying the service handler code
Implication	<ul style="list-style-type: none">▪ The adapter has to know about the data model of the wrapped domain and the component▪ The adapter transforms the data to the common model▪ The interface of the adapter must be the same as façade would be if the Domain Façade Pattern was being used▪ See also<ul style="list-style-type: none">▪ Domain Façade Guideline

Components can only interact with each other through defined services.



- Components are completely isolated from other components
- Business functionality is available only through service definitions
- Bypassing service definitions is discouraged even when the business functionality of the called component is known to be in the same domain as the calling component
- An integration layer will be used for routing service calls

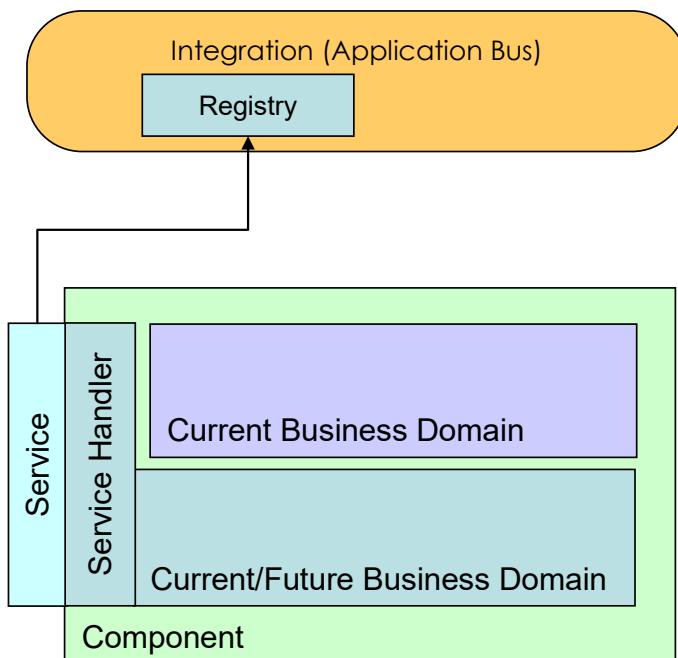
All access to interactive business functionality is through service definitions.

Disposition	Guideline
Intention	Provide a single mechanism for accessing interactive business functions
Motivation	<ul style="list-style-type: none">▪ Enable service implementations to be changed without impacting code▪ Eliminate undocumented tightly-coupled interactions
Implication	<ul style="list-style-type: none">▪ Components cannot depend on internal implementation details of other components▪ Interactions between components is designed for interactive access▪ See also<ul style="list-style-type: none">▪ Service Exposure Guideline▪ Domain Internal Exposure Guideline

All access to services will be through an integration layer

Disposition	Guideline
Intention	Isolate the discovery and routing of services
Motivation	<ul style="list-style-type: none">▪ Enable service implementations to be changed without impacting code▪ Eliminate undocumented tightly-coupled interactions
Implication	<ul style="list-style-type: none">▪ Components cannot make point-to-point calls to another component▪ All services must be published to an integration layer▪ See also<ul style="list-style-type: none">▪ Service Discovery Guideline▪ Domain Internal Exposure Guideline

A Registry is used to contain the details of service location.



- The integration layer uses a registry to find service components
- The component exposes services through a registry in the integration layer
- The registry may be a logical or physical structure in the integration layer
- Registry entries must be able to change without impacting client code

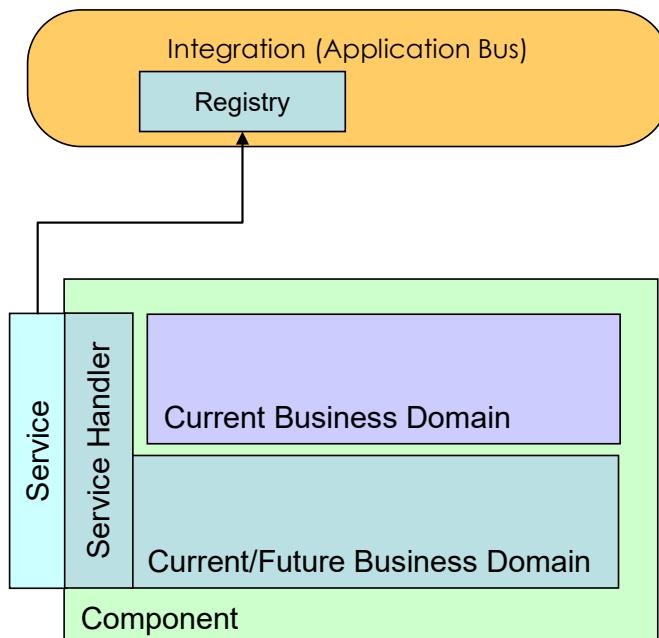
Expose services to the integration layer through a registry

Disposition	Guideline
Intention	Provide a single place where service components can be resolved
Motivation	<ul style="list-style-type: none">▪ Enable service component locations to be changed without impacting code
Implication	<ul style="list-style-type: none">▪ The technology used to implement the registry is aligned with the integration layer implementation▪ Point-to-point connectivity is discouraged▪ See also<ul style="list-style-type: none">▪ Domain Internal Exposure Pattern

Domain Internal Exposure Pattern

Pattern

The domain internal exposure pattern can be used when multiple components are implemented in a single domain.

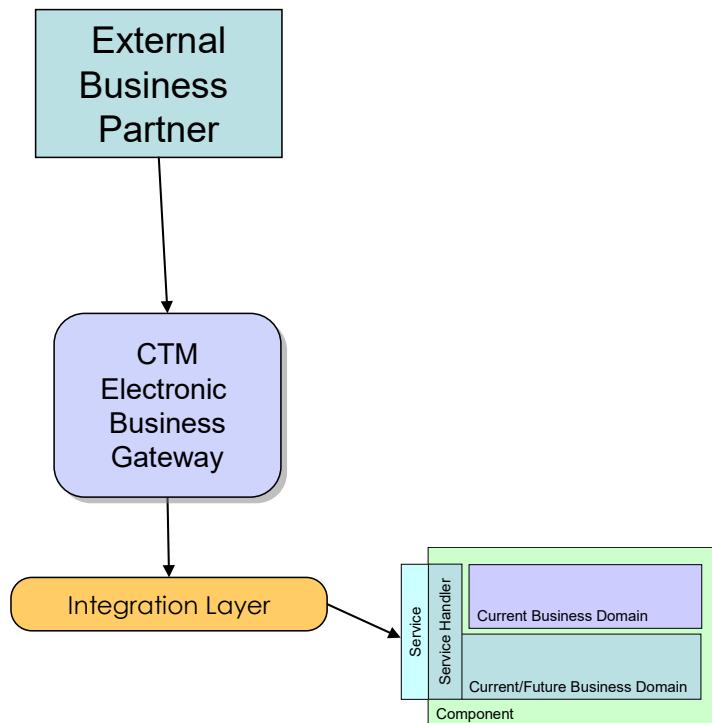


- This pattern is a specialization of the Service Exposure Pattern
- This pattern violates component encapsulation but is accommodated for transitioning existing assets
- The internal bus should be able to call through to external services
- In this version, the registry and integration layer are both implemented in the same technology platform as the rest of the domain
- The service definitions and data models are still the same as for all other services
- It is preferred that the service still be exposed externally
- The calling code cannot bypass the service definition or be aware of domain internal implementation details

Expose services through an internal integration layer for domains with multiple components

Disposition	Guideline
Intention	Balance performance with adherence to architectural standards
Motivation	<ul style="list-style-type: none">▪ Increase performance by using native calling technologies▪ Allow efficiencies while transforming existing assets
Implication	<ul style="list-style-type: none">▪ The integration layer in this case will be lightweight▪ The integration layer will be used for internal and external calls and will use an external application bus for calls to external domains▪ The calling code should not know if the integration is internal or external▪ See also<ul style="list-style-type: none">▪ Service Exposure Pattern

Services are exposed to users outside the organization through standards based services via the CTM Electronic Business Gateway.

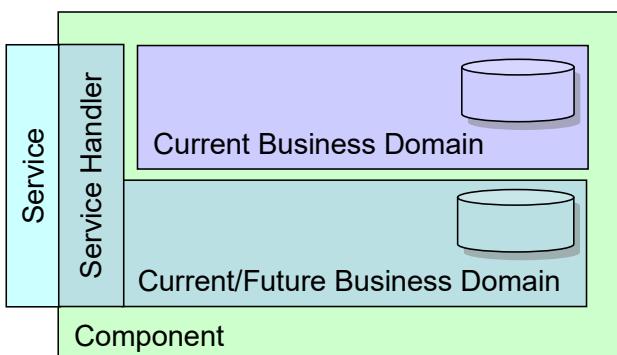


- All external user service calls will be routed through the CTM Electronic Business Gateway
- The CTM Electronic Business Gateway will use an integration layer to find BU' exposed services
- Services exposed externally will be exposed as web services except where contractually required
- External services will be based on industry standards (ACORD, NAVA, etc) except where contractually required
- In the case of non-standard interfaces or connectivity transformation to standard forms will be isolated from other code and implemented by infrastructure where possible
- Point-to-point connectivity is discouraged

Expose services to external partners as standards based services on the CTM Electronic Business Gateway.

Disposition	Guideline
Intention	Leverage industry and corporate standards
Motivation	<ul style="list-style-type: none">▪ Minimize custom technology implementations▪ Leverage CTM supported standards▪ Maximize the potential reuse by business partners
Implication	<ul style="list-style-type: none">▪ Services exposed to business partners will be available to internal users as well▪ Point-to-point and non-standard implementations are discouraged▪ See also<ul style="list-style-type: none">▪ Service Exposure Pattern

Each component is responsible for access to associated transactional or operational data within a domain.



- Access to this data is only available through services exposed by components
- Components provide access to all manipulations of the associated data stores
- The partitioning and consolidation of data across domains is handled internally to the component
- Data stores cannot be accessed across domains even within a component
- Data stores cannot be accessed across components within a single domain

Interactive access to data stores are partitioned in alignment with domains within components

Disposition	Guideline
Intention	Ensure that data is not accessed outside the control of a component
Motivation	<ul style="list-style-type: none">▪ Enhance the quality and control over data▪ Enable data model changes by ensuring isolated access points
Implication	<ul style="list-style-type: none">▪ Service calls need to be defined for all data access▪ Service calls may need to support transactional semantics▪ Domains cannot use the transactional store of another domain directly even within a component▪ See also<ul style="list-style-type: none">▪ Batch pattern

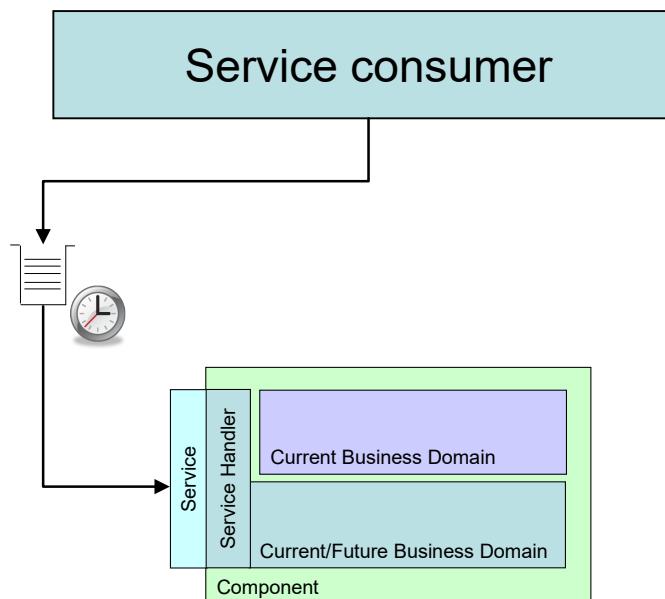
Database technology for a domain should be aligned with the development environment

Disposition	Guideline
Intention	Standardize the selection of database technologies
Motivation	<ul style="list-style-type: none">▪ Simplify the infrastructure environment▪ Leverage CTM support capabilities
Implication	<ul style="list-style-type: none">▪ The Oracle database is preferred for all non-Progress development▪ The Progress database is preferred for Progress development

Delayed Transaction Pattern

Pattern

Transactions that depend on the timing of external events may be delayed but will still be handled in an interactive manner.

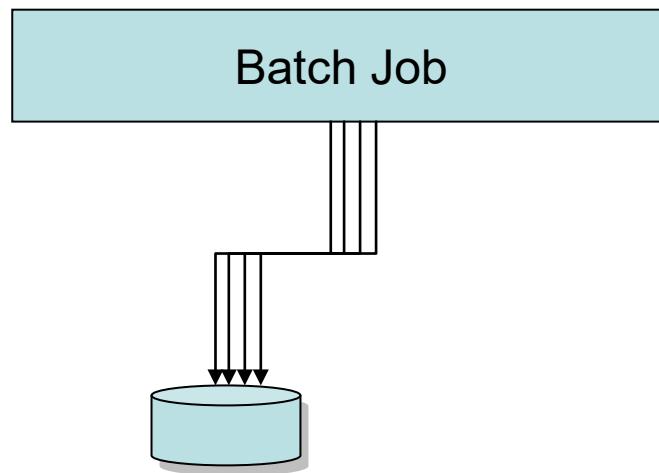


- Services will be designed to store requests in a queue and will provide a means for checking status
- Delayed transactions will utilize services that could be used interactively
- Delayed transactions will be preferred for all overnight processing that does not require access across multiple disjoint data records

Delayed Transactions will be used to handle overnight processing of individual transactions

Disposition	Guideline
Intention	Use the same service architecture for overnight processing as for interactive processing
Motivation	<ul style="list-style-type: none">▪ Consistency of business functionality▪ Compliance with business or regulatory requirements
Implication	<ul style="list-style-type: none">▪ Most batch processing will become a sequence of queue reads and interactive service calls▪ Delayed transactions should not interfere with other interactive access▪ See also<ul style="list-style-type: none">▪ Batch pattern

Jobs that require write access to multiple business records at a time will have direct access to the underlying data store.

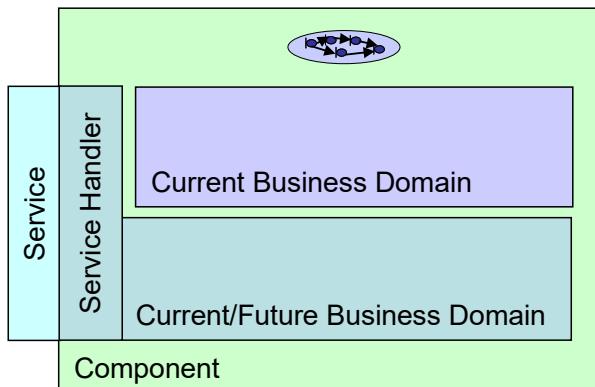


- Batch jobs will required dedicated times to run
- Batch jobs will open transactions that will potentially be long running and involve multiple records that are not logically associated with each other

Batch process should be used for write access involving multiple disjoint database records

Disposition	Guideline
Intention	Allow direct database access for bulk processing to ensure maximum performance
Motivation	<ul style="list-style-type: none">▪ Minimizing overhead of batch jobs
Implication	<ul style="list-style-type: none">▪ Batch processing will run in dedicated batch windows to avoid interfering with ETL jobs and interactive access▪ The business functionality of a batch process will not be available for reuse through a service▪ See also <p>▪ Delayed Transaction pattern</p>

Business functionality which coordinates multiple services should be isolated from other business logic.



- Isolating the process minimizes the amount of code that has to be examined or modified in order to support a business process change
- Processes may be handled by a component of the integration layer or may be embedded within a domain
- Process configuration should be designed to be changed easily
- Processes should get all business functionality through services

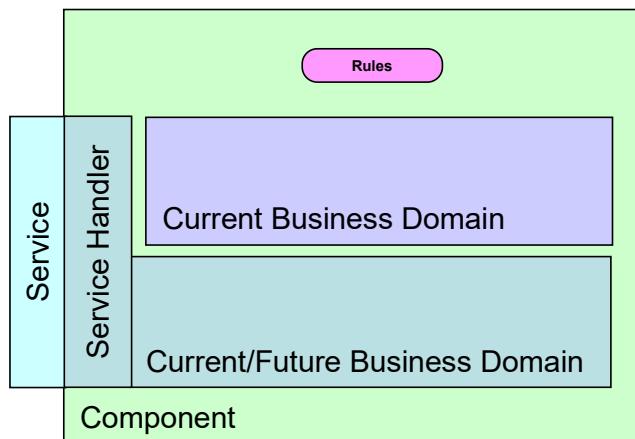
Isolate business process flow from business function logic

Disposition	Guideline
Intention	Create an distinction between process and function
Motivation	<ul style="list-style-type: none">▪ Enhance ability to adapt to business process changes
Implication	<ul style="list-style-type: none">▪ Business process may be expressed in a different technology than functional logic▪ Criteria to determine the boundaries between functions and process need to be defined▪ Organizational skills in process modeling need to be enhanced

Process engines are an emerging technology

Disposition	Guideline
Intention	State what tool to develop process code in
Motivation	<ul style="list-style-type: none">▪ Multiple approaches to implementing process models exist▪ Developers need guidance to determine proper tool
Implication	<ul style="list-style-type: none">▪ Developers should currently use the same languages for function and process code▪ Process modeling languages (BPEL, etc) are not to be used for production▪ The Architecture Office will continue to investigate process engines

Complex business logic should be described as a set of rules and isolated from other code.



- Isolating the rules minimizes the amount of code that has to be examined or modified in order to support a business function change
- Rules may be implemented in a number of technologies, including traditional programming models, data driven programming, and rules languages
- Rules should be designed to be changed easily
- Rules may be expressed as services through a component or may be used by service implementations

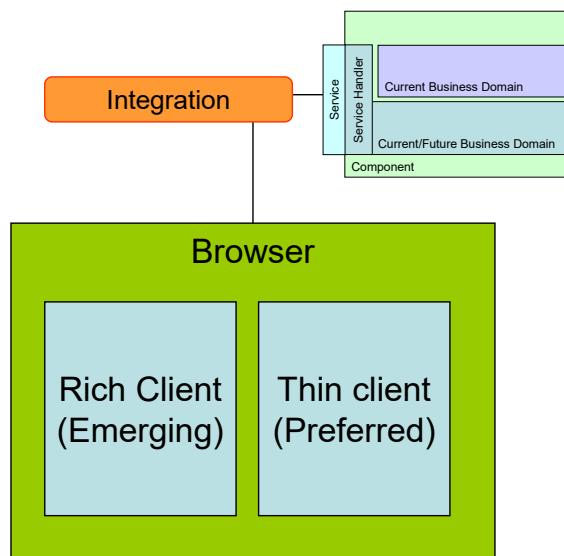
Express complex business logic as a set of configurable rules

Disposition	Guideline
Intention	Make complex logic more maintainable
Motivation	<ul style="list-style-type: none">▪ Business logic can easily become unreadable when expressed as code▪ Business views requirements as having parameters that can be modified to change result values
Implication	<ul style="list-style-type: none">▪ Criteria to determine which business logic should be implemented as rules need to be defined▪ Access to the business rules should be encapsulated as a single interface within a component▪ Standards around implements rules need to be defined▪ Organizational skills in rules need to be enhanced

Rules languages are an emerging technology

Disposition	Guideline
Intention	State what tool to develop rules code in
Motivation	<ul style="list-style-type: none">▪ Multiple approaches to implementing rules exist▪ Developers need guidance to determine proper tool
Implication	<ul style="list-style-type: none">▪ It is preferred to implement rules in the same language as other business logic▪ It is preferred to store decisioning criteria for the rules in dedicated data stores▪ Rules languages (Blaze, etc) are not to be used for production▪ The Architecture Office will continue to investigate rules languages

Browser-based interactions are the preferred graphical user interface interaction model.



- No business logic should exist in the presentation code. The business logic should be completely contained in services exposed by one or more components
- A business delegate layer are preferred in the presentation code to handle screen-to-screen transitions. These business delegate layers will coordinate the invocation of component based services
- Simple field validations (such as numeric format and string lengths) are preferred to be in presentation code rather than requiring service calls. The field restrictions should be retrieved from services where possible
- The browser-based interaction accesses the component through the integration layer

Browsers are the preferred client for graphical user interfaces

Disposition	Guideline
Intention	Set the standard for graphic user interface design
Motivation	<ul style="list-style-type: none">▪ Browsers are ubiquitous▪ Desktop installations are not required▪ Presentation can be common internal and external to the company
Implication	<ul style="list-style-type: none">▪ Standards around browser presentation need to be established

Thin client browser interfaces implements using J2EE/Struts is the preferred model

Disposition	Guideline
Intention	State the preferred graphic user interface tooling
Motivation	<ul style="list-style-type: none">▪ Standardize GUI development
Implication	<ul style="list-style-type: none">▪ The Model/View/Controller model should be enforced▪ Services exposed from components will implement business logic

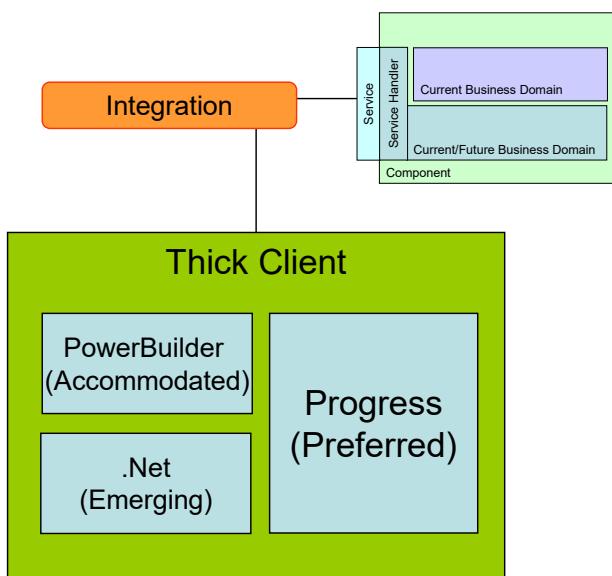
Thin client browser interfaces using Progress WebSpeed are accommodated

Disposition	Guideline
Intention	State the preferred graphic user interface tooling
Motivation	<ul style="list-style-type: none">▪ Standardize GUI development
Implication	<ul style="list-style-type: none">▪ The Model/View/Controller pattern should be followed▪ Care must be taken to ensure no business logic (beyond simple screen validations) exists in the presentation layer

Rich client browser interfaces are emerging

Disposition	Guideline
Intention	State the preferred graphic user interface tooling
Motivation	<ul style="list-style-type: none">▪ Standardize GUI development
Implication	<ul style="list-style-type: none">▪ Rich clients should not be used for production use without Architecture Office approval▪ Thin and Thick clients are alternatives to consider

Thick Client interactions should be created only where complex user interfaces (including multiple screens, complex graphics or the like) are required.



- No business logic should exist in the presentation code. The business logic should be completely contained in services exposed by one or more components
- A business delegate layer are preferred in the presentation code to handle screen-to-screen transitions. These business delegate layers will coordinate the invocation of component based services
- Simple field validations (such as numeric format and string lengths) are preferred to be in presentation code rather than requiring service calls. The field restrictions should be retrieved from services where possible
- The Thick-Client view accesses the component through the integration layer

Progress is the preferred Thick Client development environment

Disposition	Guideline
Intention	State the preferred graphic user interface tooling
Motivation	<ul style="list-style-type: none">▪ Standardize GUI development
Implication	<ul style="list-style-type: none">▪ The Model/View/Controller pattern must be used▪ Services exposed from components will implement business logic

Application Architecture Components (1/3)

The current list of defined components and their function is below.

- Accounting – BU & Corporate Financials & Tax
- Activities – Interactions with key external stakeholders
- Commissions – Financial Professional commissions
- Content Management – Dynamic web content and document templates
- Contract – Core contract processing: names & address, etc.
- Contract Financial – Contract financial processing: subpays, surrenders, etc.
- Correspondence – Tax and other form letters
- Document Management – Storing and scanning of documents
- Fund Trades – Money movement in Prudential accounts
- Hedging – Financial Risk Management
- Human Resources – Payroll, etc

Application Architecture Components (2/3)

The current list of defined components and their function is below.

- **Illustrations** – Projected financial return illustrations
- **Investment** – Investment vehicle selection and management
- **Knowledge** – Training materials
- **Licensing and Appointment** – Agent licensing
- **Marketing** – Brand & product positioning and materials
- **Money In** – Processing of checks and EFTs
- **Money Movement** – Moving money between sub accounts
- **Party** – Stakeholder information: name, address, etc
- **Payments** – Money out; check writing and bank transfers
- **Plan** – Group account processing
- **Product Pricing** – Set up product pricing structure
- **Product** – Design new or updated products
- **Product Performance** – Evaluate current or historical performance

Application Architecture Components (3/3)

The current list of defined components and their function is below.

- **Reporting** – Analytics and reporting
- **Reserves** – Setting aside money to meet obligations
- **Sales Compensation** – Compensation for wholesalers and SAMs
- **Sales Planning** – Strategy for calling on financial planners
- **Security** – Authorizations, entitlements, etc
- **Slippage** – Trades pending corrections
- **Suitability** – Determining suitability for new money
- **Suspense** – Money for accounts in transition
- **Tax** – Reporting contract tax events
- **Valuations** – Determining contract value
- **Value Added Services** – Extra services for contract holders or FPs

Agenda

1 RA Vision, Principles, Policies, and Guidelines

2 RA Blueprinting - Business Architecture

3 RA Blueprinting – Application Architecture

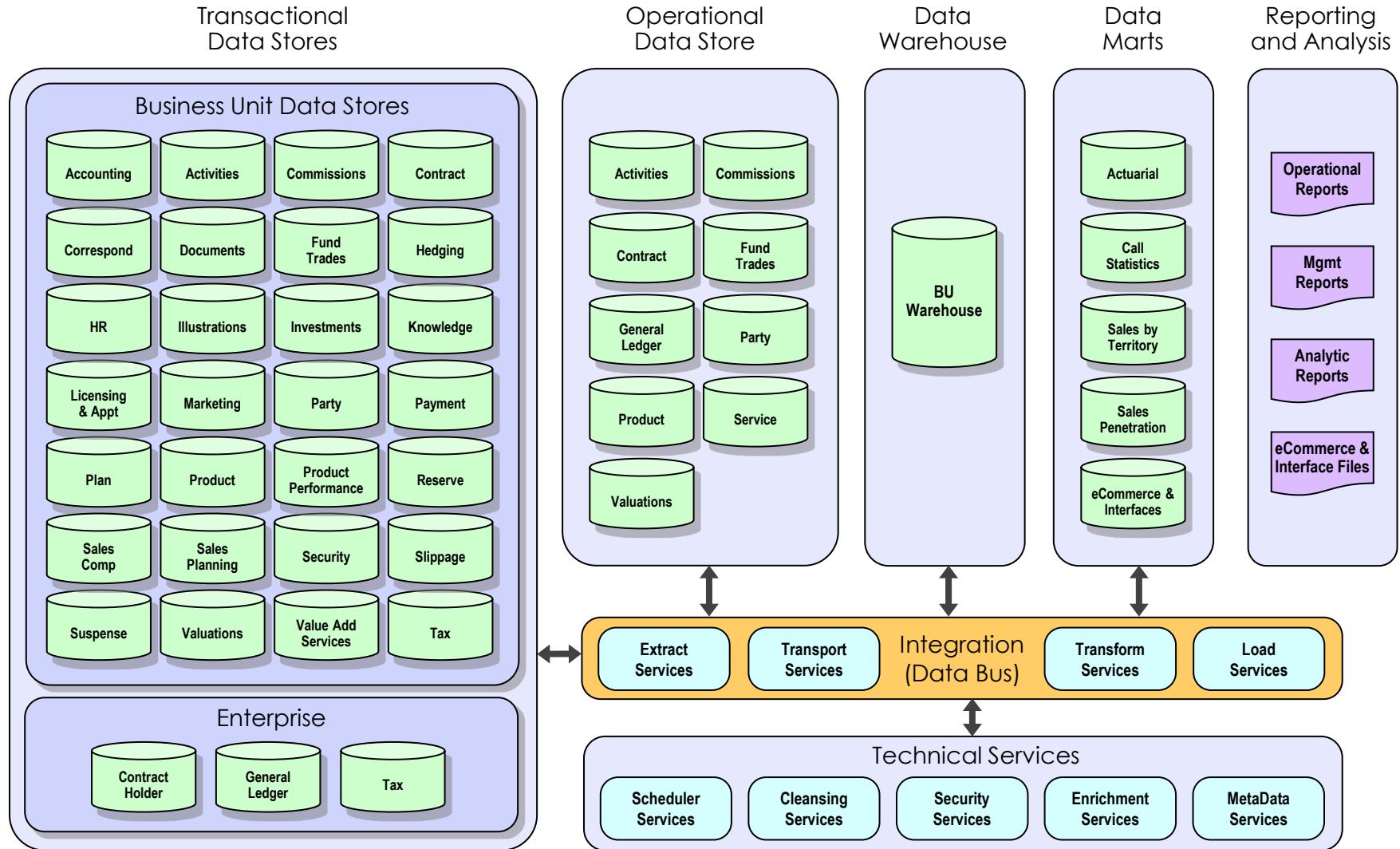
4 RA Blueprinting – Information Architecture

5 RA Blueprinting – Technical Infra. Architecture

6 Appendix A: Business Architecture Details

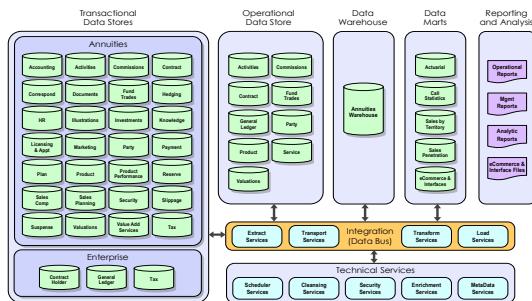


Information Architecture Diagram



Information Architecture Description

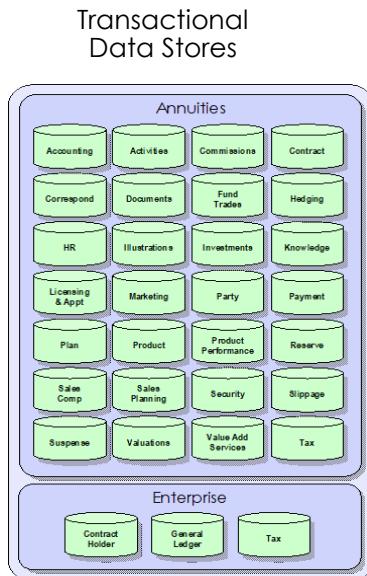
The Information Architecture addresses the information and data aspects of the Business Unit Reference Architecture. This architecture include transactional, operational, and analytic data in both structured and unstructured formats. It aligns with the Application Architecture data, integration, and technical services layers and is supported by the nodes in the infrastructure architecture.



- Transactional data stores contains the day to day data used to run the business. Information is created and updated by the business unit's set of applications and systems.
- Operational data stores contain the consolidated data used for inquiry and reporting. Data from the transactional stores is consolidated and loaded into the ODS.
- The data warehouse is a repository for storing integrated information related to the business. Information is loaded into the warehouse from the operational data and transactional stores. Information is cleansed and organized for efficient query and analysis.
- Data marts are subsets of the data in the data warehouse. They are consolidated and summarized to provide efficient reporting and analysis for a specific set of user requirements
- Reporting and analysis represents the tools that enable query and analysis of the data in the various data stores.
- Integration provides a loosely coupled connection between source and target data store enabling extract, transport, transformation and loading of data within and between layers of the information architecture.
- Technical services provide the more advanced capabilities for data cleansing, enrichment, scheduling movement of data, and meta data management related to extract transform and load of data.

Transactional Data Stores

Transactional data stores contains the day to day data used to run the business. Information is created and updated by the set of business unit's applications and systems.



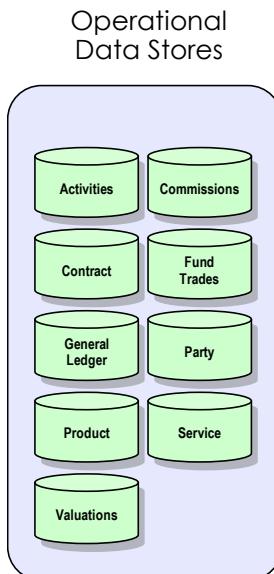
- The transactional layer contains numerous data stores.
- Typically there is a single data store per subject area. In a small number of cases, multiple instances of a transactional store exist. Multiple instances of the same data store occur when more than one existing application performs the same function. e.g. multiple contract systems exist each one containing its own data store. Multiple instances of the same data store exist for Party, Contract, Commissions, Fund Trades and Product.
- Data in transactional stores are created and updated by the functional components defined in the application architecture, data is not loaded through ETL tooling like the other information store layers.
- The transactional layer also contains standard reference data, business rules, and security authorization rules.
- Transactional data stores can contain structured and non structured data. The Documents data store provides a repository for all unstructured information including:
 - PDF contract application, scanned documents, correspondence.
 - Web trace, mail & chat archive, service call audio archive
 - Contract documents, historical report files
- Most of the information stores are owned by the business unit however, Contract Holder (CIS), Tax and General Ledger are owned by corporate. BU feeds data to these corporate stores.

Historical Information	Update Frequency	Level of Summarization	Transaction Support	Structure and Organization
History based on needs of application and compliance	Real time	None	Allows all CRUD functions	Application centric Structured for performance, storage, and maintenance

Operational Data Stores

Operational data stores contain the consolidated data used for inquiry and reporting.

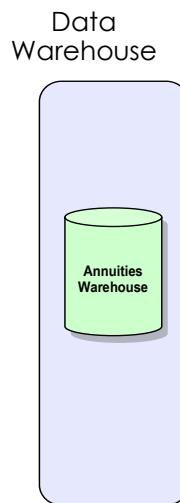
- Multiple Operational Data Stores will be created to support various subject areas.
- Data from the transactional systems is consolidated and loaded into the Operational Data Store.
- Operational Data Store characteristics are defined in the table below.



Historical Information	Update Frequency	Level of Summarization	Transaction Support	Structure and Organization
Can include history if the transactional store does not maintain history and if consumers of the ODS require history	Near real time	None	Read only	Optimized for read access across multiple applications. Structured for enterprise (BU or corporate) view

Data Warehouse

The Data Warehouse is a repository for storing integrated information related to the business. Information is loaded into the warehouse from the operational data. Information is cleansed and organized for efficient query and analysis.



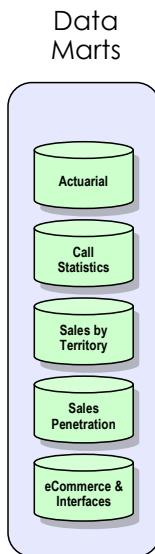
- A single BU data warehouse containing multiple domains is implemented to support reporting and analysis.
- Domains include Transaction, AUM, GAAP, Touch, Service.
- Data flows from the ODS and is cleansed and organized before loading in the warehouse (e.g. Transaction and AUM information would be extracted from contract in the ODS and organized in the warehouse for efficient query and analysis).
- Data warehouse characteristics are described in the table below.

Historical Information	Update Frequency	Level of Summarization	Transaction Support	Structure and Organization
Full Historical data, if needed by the domain, including reference and transactional history	As needed by the consumers.	The guideline is for no summarization or roll ups	Read Only	Structured for reporting

Data Marts

Data marts are subsets of the data in the data warehouse consolidated and summarized to provide efficient reporting and analysis for a specific set of user requirements.

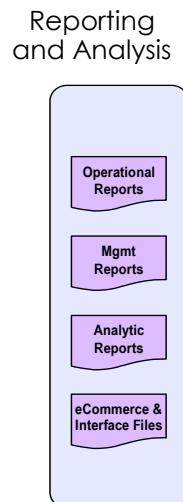
- Multiple data marts will be created to support the various reporting and analysis requirements.
- Data flows from the Operational Data Store to the warehouse where it is extracted and loaded into the required data marts.
- Data mart characteristics are described in the table below.



Historical Information	Update Frequency	Level of Summarization	Transaction Support	Structure and Organization
Point in time	As needed by the consumers	Summarized, rolled up to meet requirements of consumer	Read only	For optimized Reporting

Reporting and Analysis

The Reporting and Analysis layer contain the tools that enable query and analysis of the data in the various data stores.



- Standard tooling such as Cognos will be used for developing reports.
- The following list identifies some of the report types that are required:
 - Operational Reports
 - Management Reports
 - Analytic Reports
 - eCommerce & Extract Files

Integration Layer

The Integration layer provides a loosely coupled connection between source and target data stores enabling extract, transport, transformation and loading of data within and between layers or the information architecture.



- The Integration Layer provides basic data services for:
 - Extract Services
 - Transport Services
 - Transform Services
 - Load Services

Technical Services Layer

Technical services provides the capabilities required to enable the information architecture. Technical services also host common services that support both business and technology capabilities.

- The following technical services support the information architecture:
 - Scheduler Services used to control execution based on time and event occurrences.
 - Cleansing Services used to improve the accuracy and quality of data.
 - Security Services used for basic authentication and authorization of data.
 - Enrichment Services used for complex transformation or enhancement to data.
 - MetaData Services used to setup and maintain data definitions and mapping.



Information Architecture Data Store Layers Summary

	Transactional Data Stores	Operational Data Stores	Data Warehouse	Data Marts
Historical Information	History based on needs of application and compliance	Can include history if the transactional store does not maintain history and if consumers of the ODS require history	Full Historical data, if needed by the domain, including reference and transactional history	Point in time
Update Frequency	Real Time	Near Real time	As needed by the consumers	As needed. by the consumers
Level of Summarization	None	None	The guideline is for no summarization or roll ups	Summarized, rolled up to meet requirements of consumer
Transaction Support	Allows all CRUD functions	Read only	Read Only	Read only
Structure and Organization	Application centric. Structured for application performance, storage and maintaining the data	Optimized for read access across multiple applications. Structured for enterprise (BU or corporate) view	Structured for reporting	For optimized reporting

Policies / Guidelines

Information

- A BU information model will be used to assure a common and consistent logical definition of data across the BU organization
- Data should be treated as a corporate asset
- Data should not be replicated within the same information layer
- Access to shared data, data placement, data management and data replication should be based on a set of BU-wide standards for data access and placement
- Referential Integrity is an attribute of the data layer not of the application layer
- Data stores will meet the service level agreements of each layer of the application architecture
- Data in a Transactional Data Store that is needed in the Data Warehouse is stored first in the Operational Data Store
- The architecture shall support transport of data between and within layers at various levels of granularity and frequency
- Oracle should be used as the database management system (DBMS) for new development

A BU information model will be used to assure a common and consistent logical definition of data across the organization

Disposition	Policy
Intention	Common vocabulary describing BU information as well as enable easier and consistent exchange of information between systems, applications, and components
Motivation	<ul style="list-style-type: none">▪ Provide a consistent definition of all attributes▪ Provide a common understanding of relationships between data and systems▪ Common business components can be developed more efficiently using a common information model▪ Helps to provide correct and consistent information across the business (by using common agreed upon business definitions)▪ Provides the base for a common business language across business units▪ Improved ability to provide BU-wide business intelligence functionality▪ Reduced IT development effort
Implication	<ul style="list-style-type: none">▪ Need to communicate model and definitions to applications, systems and new projects▪ Individual development groups need to agree to use common definitions for the data they maintain and use▪ Need to define rules for conformance and mapping to the model; rules can be based on existing systems, new systems and purchased software packages; rules will impact selecting and purchasing of applications as they come with their own data model▪ Having a common model does not require every application to use every element available▪ Multiple data stores can map to the same common information model but not all sources will contain a complete set of attributes▪ Applications need to identify if the complete set of attributes required by the application exist in the ODS or transactional stores▪ Applications must evaluate the source of required data based on the availability of the data in the transactional or operational data stores

Data should be treated like a corporate asset

Disposition	Guideline
Intention	<p>Information is a BU-wide business asset, and it should be captured, stored and managed in a way that will allow appropriate levels of sharing across the organization based on business need and security rules</p>
Motivation	<ul style="list-style-type: none">▪ Ensures consistent view and use of information throughout the organization▪ Improves the coordination and integration between applications and systems▪ Incorrect or inconsistent data inhibits decision making▪ Incorrect or inconsistent data sets are difficult to combine in ways that enable future business functions
Implication	<ul style="list-style-type: none">▪ Information of value must be identified and rules for its usage and sharing must be established▪ Need to establish the role of Information Owners who manage 'their' information▪ Data management must be unified▪ Data owners must be identified▪ Data quality must be auditable▪ Data warehouses should be developed to facilitate information availability for decision making.▪ Changes in application logic may be needed to enable information (data) usage and sharing▪ Data Encryption for data during movement and while at rest must be considered▪ Obfuscation of data is required by corporate policy when production data is used in non production environments e.g. test

Data should not be replicated within the same information layer

Disposition	Guideline
Intention	Prevent replication of data and business rules as well as reuse existing business components
Motivation	<ul style="list-style-type: none">▪ Need for single consistent view of data▪ Support the application architecture principal of data encapsulation within a component▪ Reduce the ability for a component to access another components data directly
Implication	<ul style="list-style-type: none">▪ Additional data may be needed in the Operational Data Store to meet the availability needs of the application layer and prevent replication

Access to shared data, data placement, data management and data replication should be based on a set of BU-wide standards for data access and placement

Disposition	Guideline
Intention	To improve data reuse and reduce replication within the same layer
Motivation	<ul style="list-style-type: none">▪ Increased accessibility and share-ability of data▪ Ensure data accuracy, integrity, currency, and security▪ Avoid unnecessary data redundancy▪ Simplify application development and maintenance
Implication	<ul style="list-style-type: none">▪ A consistent set of rules will need to be defined and enforced for determining the placement of distributed data▪ Application access to shared data will use standardized interfaces (a link between application and information architecture)▪ Need to design systems, processes and physical databases that provide the capability for continuous availability of shared data▪ Need to provide management tools and infrastructure that deliver data monitoring and management capabilities enabling continuous availability of shared data▪ Existing databases may need to be restructured and existing applications may need to be redesigned to align with the new information standards

Referential integrity is an attribute of the data layer, not of the application layer

Disposition	Policy
Intention	Assure referential integrity across the data regardless of the application
Motivation	<ul style="list-style-type: none">▪ Increase separation between application and data▪ Application layers can change but the data relationships typically stay the same▪ Reduce non business logic coding in the application
Implication	<ul style="list-style-type: none">▪ Application layer must still understand the underlying data relationships

Data stores will meet the service level agreements of each layer of the application architecture

Disposition	Policy
Intention	Data should be available as needed by the application, and thus the application architecture should not need to be concerned with availability of data
Motivation	<ul style="list-style-type: none">▪ Reduce risk of business impact due to unavailability of the data stores
Implication	<ul style="list-style-type: none">▪ Each data store must be accessible while batch type work is in progress▪ Infrastructure may need to be updated to meet the availability requirements of the applications and data

Data in a Transactional Data Store that is needed in the Data Warehouse is stored first in the Operational Data Store

Disposition	Policy
Intention	The Operational Data store is the staging area for the warehouse
Motivation	<ul style="list-style-type: none">▪ Provide a control point for normalization, consolidation, and validation of the data being loaded in the warehouse
Implication	<ul style="list-style-type: none">▪ Excess data not required by Operational Data Store consumers may be stored in the Operational Data Store▪ May take longer for data to appear in the warehouse

The architecture shall support movement of data between and within layers at various levels of granularity and frequency

Disposition	Guideline
Intention	Provide data to consumers on a timely basis
Motivation	<ul style="list-style-type: none">▪ Data is needed by consumers in various data stores at each layer in the architecture at various frequencies
Implication	<ul style="list-style-type: none">▪ Infrastructure must enable real-time and scheduled ETL capabilities▪ Data stores must support publishing changes on an as occurs basis

Oracle will be used as the database management system (DBMS)

Disposition	Policy
Intention	Standardize DBMS usage
Motivation	<ul style="list-style-type: none">▪ Reduce the number of DBMS systems in use across BU▪ Be able to take advantage of CTM utility stacks supporting database deployment
Implication	<ul style="list-style-type: none">▪ The Progress database will continue to be accommodated in support of existing Progress based applications▪ Use of other database systems is discouraged▪ Use of non Oracle/Progress database will continue until replacement is justified

Information

Transactional Data Store

- » Transactional updates are only to be performed at the transactional store

Operational Data Store

- » The ODS should contain consolidated and or normalized data
- » Master inquiry searches are done against the operational data store while detailed information inquiries are performed against the transactional data layer

Data Warehouse

- » Data stored in the warehouse should be sourced from the operational data store

Data Marts

- » Data for the Data Marts should be sourced from the warehouse

Reporting and Analysis

- » Reporting should be done at the ODS, Warehouse and Data Mart layers

Integration

- » Movement of data between stores should be done using appropriate tooling

Transactional updates are only to be performed against Transactional Data Stores and not other data stores

Disposition	Policy
Intention	<ul style="list-style-type: none">▪ Provide a single source of data creation, maintenance and deletion
Motivation	<ul style="list-style-type: none">▪ Create a unidirectional flow of information through all layers
Implication	<ul style="list-style-type: none">▪ Changes may take longer to appear in the non transactional layers▪ The Operational Data Store, Data Warehouse, and Data Mart are only used for reading information

The Operational Data Store should contain consolidated and or normalized data

Disposition	Guideline
Intention	Only consolidated or normalized data required by consumers must be available in the ODS, and summarized data is not to be stored in the ODS
Motivation	<ul style="list-style-type: none">▪ Only need to replicate data if needed by consumers
Implication	<ul style="list-style-type: none">▪ Each data element must be analyzed to determine if it belongs in the ODS▪ Not all data in the transactional layer will be in the ODS

Application master inquiry searches are done against the operational data store while application detailed information inquiries are performed against the transactional data layer

Disposition	Guideline
Intention	Use consolidated or normalize data structures when performing data inquiry, and use detailed information in the transactional layer prior to updating data
Motivation	<ul style="list-style-type: none">▪ Provide a federated search capability for applications
Implication	<ul style="list-style-type: none">▪ Not all attributes are available in the federated search results▪ Reporting should perform detailed inquiries against the ODS when the data is available in the ODS

Data stored in the warehouse should be sourced from the operational data store

Disposition	Guideline
Intention	<p>Data stored in the warehouse should start in the transactional layer, be consolidated and normalized in the ODS and then moved to the warehouse</p>
Motivation	<ul style="list-style-type: none">▪ Desire not to replicate data just for sake of putting in the warehouse▪ Only want to replicate data in the ODS if consolidation, normalization or access by ODS consumers is needed▪ Need to provide historic information in the warehouse when the information is not stored in the source systems
Implication	<ul style="list-style-type: none">▪ Transactional data may be loaded into the ODS then independently into the warehouse requiring coordination to assure consistent data is loaded into both data stores▪ External or reference data may go directly to the warehouse bypassing the ODS

Data for the Data Marts should be sourced from the Data Warehouse

Disposition	Guideline
Intention	Use the most efficient source of data for the data mart
Motivation	<ul style="list-style-type: none">▪ Warehouse data is generally consolidated, normalized, cleansed and therefore the most efficient source▪ Summarized data may be provided by external partners for enhanced reporting needs
Implication	<ul style="list-style-type: none">▪ Some data required in data marts may not be available in the warehouse▪ Some data provided by external parties might be required in data marts

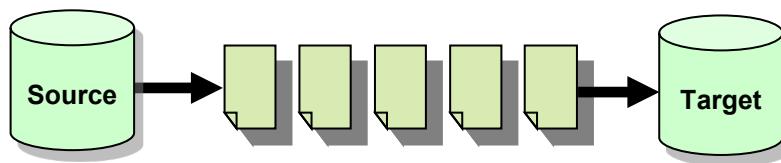
Reporting should be done against the Operational Data Stores, Data Warehouse, or Data Marts

Disposition	Guideline
Intention	<ul style="list-style-type: none">▪ Use the ODS for real-time reports when near real-time data is available otherwise use the transactional stores, use the warehouse for detailed, historical or end of day reports, and use marts for summarized reporting
Motivation	<ul style="list-style-type: none">▪ Reporting should be performed at the most efficient layer of the information architecture
Implication	<ul style="list-style-type: none">▪ Repeated summarized requests for data may generate requirements for additional data marts to be created▪ Repeated requests for detailed reports against the ODS or transactional layers may indicate the need for additional data in the data warehouse layer

Transport of data between stores should be done using appropriate tooling

Disposition	Guideline
Intention	<ul style="list-style-type: none">A standard approach for ETL of data between and within layers should be followed across the organization
Motivation	<ul style="list-style-type: none">Tools can make use of the BU information model to rapidly develop, deploy and maintain ETL capabilities for all types of data<ul style="list-style-type: none">Generating flat files for sending to partnersReplicating data between data databases and layersEnable creation of reusable mapping and transformation components
Implication	<ul style="list-style-type: none">No future development of one off data transport applications should take placeThe ETL tool must be capable of moving data based on two patterns:<ul style="list-style-type: none">» Near Real Time - as data is modified» Bulk - A group of data updates moved on a less frequent scheduled basis

The near real time data transport pattern is defined for situations where consumers require near real time currency.

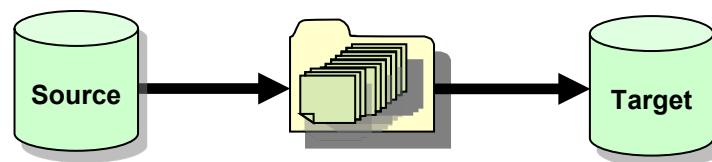


- Data is moved on a single unit of work basis (e.g. each change to a contract is moved to other data stores independent of other changes).

The near real time data transport pattern should be used when data must be replicated and the data currency needs to be close to real time

Disposition	Guideline
Intention	<ul style="list-style-type: none">A standard approach for ETL of data between and within layers should be followed across the organization
Motivation	<ul style="list-style-type: none">Provide cost effective data transport implementations that meet the currency requirements of the applicationsSupport application requirements for near real time currency of data
Implication	<ul style="list-style-type: none">No future development of one off data transport applications should take placeThe need for data currency must be considered when planning how to move data between data stores

The bulk data transport pattern moves data in groups containing multiple independent pieces of information according to a predefined schedule.



- Data updates are collected throughout a specific time period before being moved (e.g. a group of transactions that occurred over a specified period of time).
- Typically thought of as a batch movement.
- Data currency can requirements specify that the data can be less current than the prior day.

The bulk pattern should be used when data must be replicated and the data currency can be as of the prior day or less frequent

Disposition	Guideline
Intention	<ul style="list-style-type: none">▪ A standard approach for ETL of data between and within layers should be followed across the organization
Motivation	<ul style="list-style-type: none">▪ Provide cost effective data transport implementations that meet the currency requirements of the applications▪ Support application requirements for less than near real time currency of data - typically this will be a daily or less frequent movement
Implication	<ul style="list-style-type: none">▪ No future development of one off data transport applications should take place▪ The need for data currency must be considered when planning how to move data between data stores

Agenda

1 RA Vision, Principles, Policies, and Guidelines

2 RA Blueprinting - Business Architecture

3 RA Blueprinting – Application Architecture

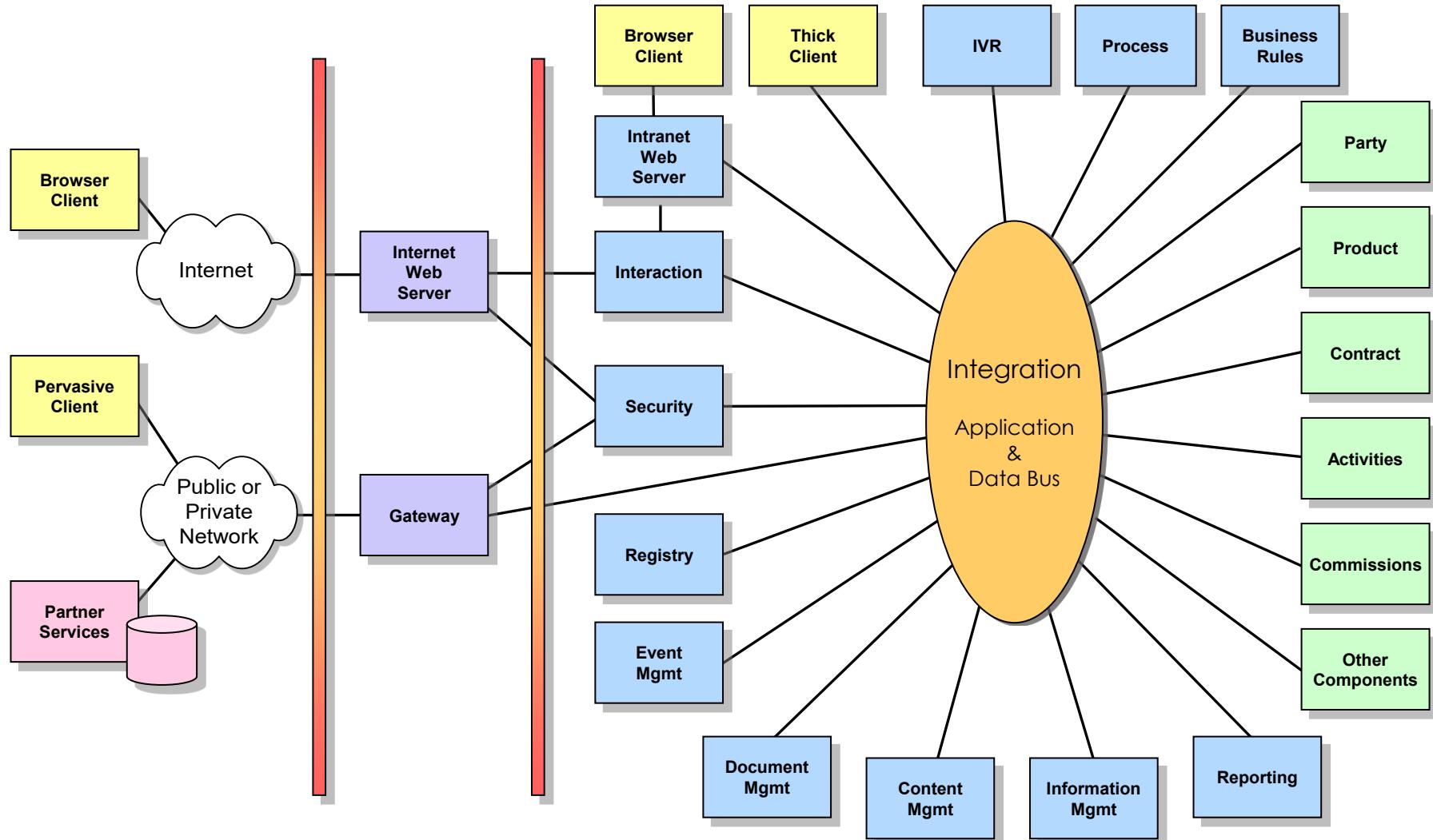
4 RA Blueprinting – Information Architecture

5 RA Blueprinting – Technical Infra. Architecture

6 Appendix A: Business Architecture Details

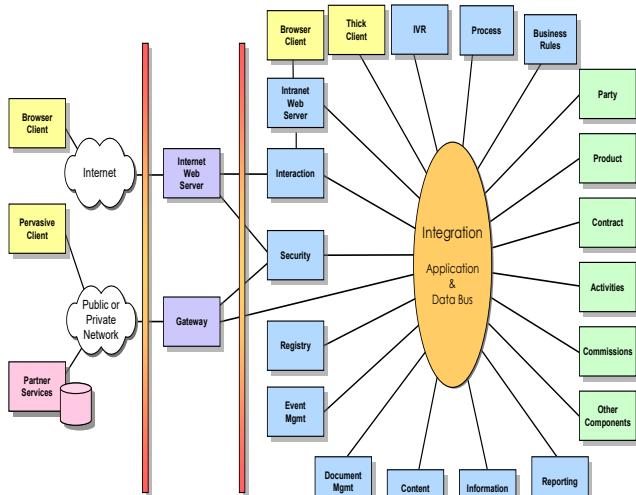


Infrastructure Architecture Diagram



Infrastructure Architecture Description

The infrastructure architecture describes the run time environments for components and other elements of the application architecture. It also provides IT services that support the application architecture.



- Each node described on the diagram aligns with and provides the run time implementation for capabilities described in the application and information architectures.
- Functional nodes provide the runtime platform for the components described in the application architecture.
- Enabling nodes provide the technology capabilities used to support BU applications.
- The integration node provides infrastructure for flexible connectivity and integration of applications and services.
- External integration nodes provide gateway capabilities to and from external parties.
- Each node is described as a set of patterns, policies, and stacks (software that implement the capability)

Infrastructure

- Business and application teams will specify the qualities of service expected of the infrastructure instead of the specific infrastructure components
- Infrastructure capabilities should be implemented with commercial off the shelf software (COTS) instead of being developed by the application team
- Infrastructure nodes should be deployed on CTM standard utility platforms and offerings where available
- Infrastructure should implement security capabilities to protect information within and beyond the Enterprise
- Infrastructure services will conform to industry standards where such standards are available
- Implement standardized IT environments that support development, testing, and implementation of applications

Business and application teams will specify the qualities of service expected of the infrastructure instead of the specific infrastructure components

Disposition	Policy
Intention	Teams should focus on the concerns specific to their domain
Motivation	<ul style="list-style-type: none">▪ The infrastructure team should be able to select the platform based on quality of service and cost of ownership independent of the applications being developed▪ It should be possible to address operational concerns independent of applications
Implication	<ul style="list-style-type: none">▪ Infrastructure should be highly available, scalable, and secure to meet or exceed all quality of service requirements▪ Business and application teams will need to specify quality of service as part of requirements

Infrastructure capabilities should be implemented with commercial off the shelf software (COTS) instead of being developed by the application team

Disposition	Guideline
Intention	Reduce non standard/custom approaches for infrastructure capabilities
Motivation	<ul style="list-style-type: none">▪ Application development teams need to focus on business functionality not infrastructure development▪ Gain efficiencies of using common tools for infrastructure capabilities
Implication	<ul style="list-style-type: none">▪ Must work with CTM to develop common capabilities▪ Must look first to CTM before attempting to develop new capabilities

Infrastructure nodes should be deployed on CTM standard utility platforms and offerings where available

Disposition	Guideline
Intention	Leverage existing corporate capabilities for application deployment with a preference for CTM standard utility platforms first, followed by CTM standard offering platforms
Motivation	<ul style="list-style-type: none">▪ Use the most cost effective infrastructure for application deployment▪ Focus more on applications and supporting the business instead of on the infrastructure▪ Prevent proliferation of multiple technologies for the same capability
Implication	<ul style="list-style-type: none">▪ Need to work with CTM to define and deploy first of a kind environments

Infrastructure should implement security capabilities to protect information within and beyond the Enterprise

Disposition	Guideline
Intention	Protect corporate and client information
Motivation	<ul style="list-style-type: none">▪ Information must be protected during exchange with partners
Implication	<ul style="list-style-type: none">▪ Various solutions including real time and batch transmissions must be implemented to secure the information▪ Security approaches should conform to industry standards to facilitate integration with partners

Infrastructure services will conform to industry standards where such standards are available

Disposition	Policy
Intention	Simplify the task of compliance with the Reference Architecture
Motivation	<ul style="list-style-type: none">▪ Maximize portability and simplify integration▪ Leverage the knowledge and support ecosystem available for industry standards
Implication	<ul style="list-style-type: none">▪ Application industry standards will need to be selected and communicated▪ Conformance to standards will be part of evaluation criteria for infrastructure product selection

Implement standardized IT environments that support development, testing, and implementation of applications

Disposition	Policy
Intention	Provide the right number of non production environments when needed and at an acceptable cost
Motivation	<ul style="list-style-type: none">▪ Improve development and testing of applications▪ Provide sufficient environments with access to dependent applications for testing purposes
Implication	<ul style="list-style-type: none">▪ IT operations need to set up sufficient and consistent development, QA, UAT, and staging environments for all applications▪ Providing computing environments that support the testing for multiple concurrent IT projects▪ Virtualization and provisioning may be needed to support multiple environments

Thick Client Node

The Thick Client node Provides a platform for applications that typically have local processing and a high function user interface.

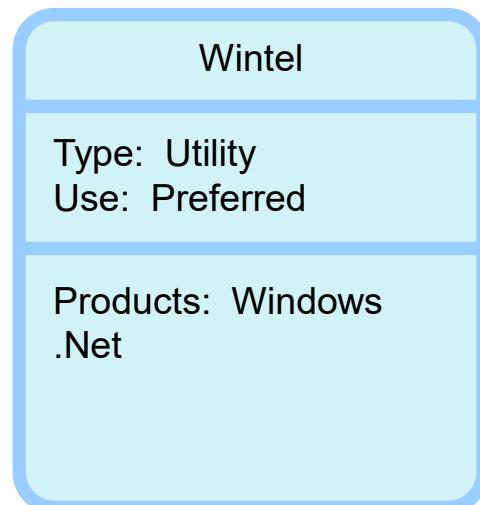


- Application Architecture Level Supported:
 - **Interaction**
- Major capabilities: **High function user interface**
- NFR: **Available during normal business hours**
- Implementation Patterns:
 - **Windows Thick Client**

The windows thick client pattern provides a platform for delivering full function user interfaces for BU applications.

Use Status: Preferred

Deployment Stack: Wintel



The windows thick client pattern should be used when a standard thin client interface does not provide sufficient capability

Disposition	Guideline
Intention	Use the Windows Thick Client pattern only when required.
Motivation	<ul style="list-style-type: none">Some application interfaces need more capability than is provided by a standard thin client interface.
Implication	<ul style="list-style-type: none">Thick client applications will need to be designed to use web services for access to functional components.

Gateway Node

The Gateway node provides a common approach for headless exchange of information between Prudential and external parties.



- Application Architecture Level Supported:
 - **Electronic Business Gateway**
- Major capabilities: **FTP using file mover; Web Services Gateway** for communicating with external parties; Both internet or private line connectivity is supported; Encryption, digital signature, and logging for both inbound/outbound web service requests
- NFR: **24x7 availability, easily scalable, recoverability, guaranteed delivery, only once delivery**
- Implementation Patterns:
 - **Web Services Gateway**
 - **Gateway File Mover**
 - **Private Line FTP**

The web services gateway pattern provides a shared tool for exchange of information between the BU and its partners. This pattern implements a bi directional flow of information and is implemented using a web services protocol.

Use Status: Preferred

Deployment Stack: Web Services Gateway



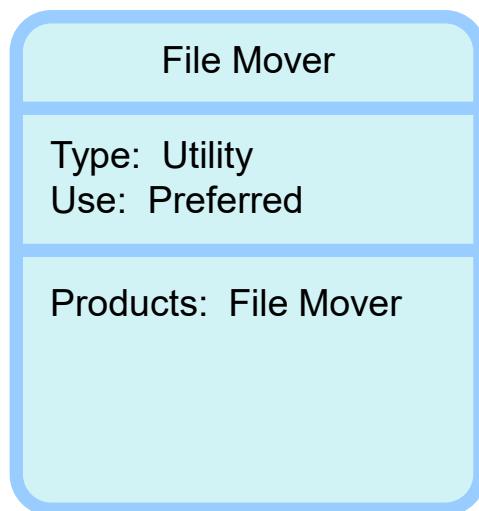
The Web Services gateway pattern should be used for integrations where the partner supports web services

Disposition	Guideline
Intention	Begin using web services instead of proprietary formats for integration
Motivation	<ul style="list-style-type: none">▪ Move toward web services for integration▪ Follow a single integration strategy for web services enabled partners
Implication	<ul style="list-style-type: none">▪ Need to work with partners to gain acceptance on use of web services

The gateway file mover pattern enables pushing information out to external partners. It is only used where the partner does not support the web services pattern. Encryption is provided through a set of public/private keys. This is the preferred method for bulk data movement.

Status: Preferred

Deployment Stack: File Mover



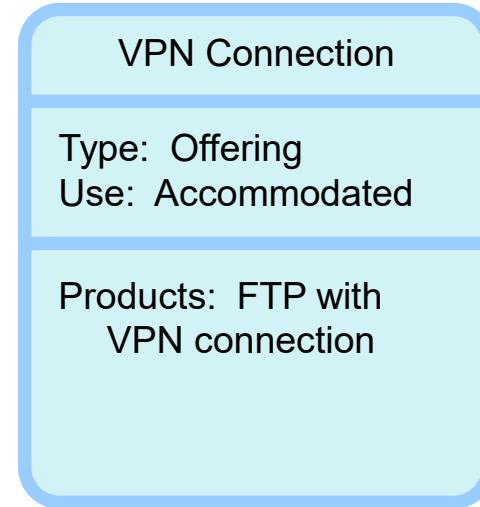
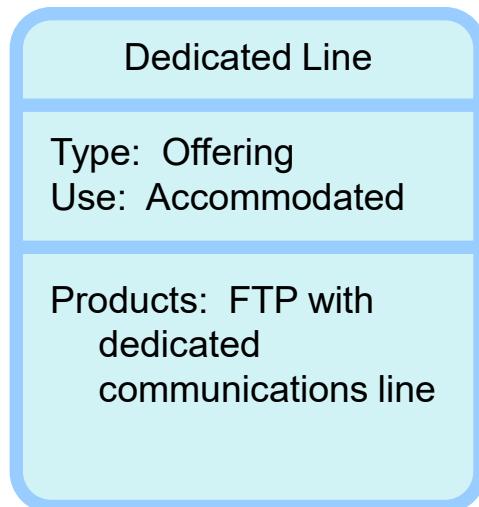
Use existing file mover for new integrations where the partner does not support web services

Disposition	Guideline
Intention	Continue to support partners who do not use web services for integration
Motivation	<ul style="list-style-type: none">Support information exchange from partners who do not support web servicesinformation exchange
Implication	

The private line FTP pattern enables pushing information out to external partners who cannot use either of the prior patterns. This pattern requires use of a dedicated communication line. Accommodated for use with clients that can't use Web services or File Mover or large files that cannot be handled by File Mover.

Status: **Discouraged**

Deployment Stack: **Dedicated Line, VPN Connection**



Event Management Node

The Event Management node provides a services based approach for managing business events.

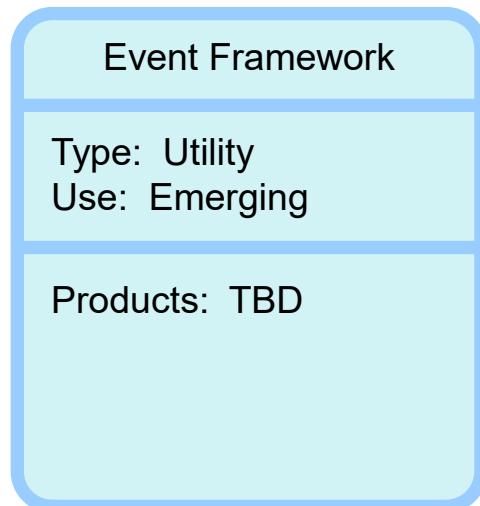


- Application Architecture Level Supported:
 - **Technical Services – Event Management**
- Major capabilities: **Creating and handling business events including logging, alerting, and reporting**
- NFR: **24x7 availability, easily scalable, recoverability, low impact**
- Implementation Patterns:
 - **Event Framework**
 - **Custom Events**

The event framework pattern will leverage industry standards to provide a common metric and monitoring infrastructure that can be consumed by all applications.

Status: **Emerging**

Deployment Stack: **Event Framework**



New components should use the event framework when not constrained by the structure of existing events

Disposition	Guideline
Intention	Implement common metric and monitoring infrastructure that can be leveraged by all applications and systems
Motivation	<ul style="list-style-type: none">▪ Move toward an environment where systems are architected "management-ready" for out-of-the-box integration with Prudential's infrastructure▪ Minimize the amount of metric and monitoring effort required by the application teams
Implication	<ul style="list-style-type: none">▪ For a period of time event management information will be stored in multiple locations▪ Reporting and analysis will need to look in both locations for a complete view▪ Do not rework the existing event scheme unless otherwise justified

In the absence of a common framework for event management, most applications have implemented custom methods for handling events. The custom approaches may include writing to log files, broadcasting messages or logging to a database. A set of tools is also developed to process the events and notify appropriate resources to take action. Typically this capability is deployed in the stack supporting the base component.

Status: **Accommodated**

Deployment Stack: **Custom Events**



Custom events should be generated only when enhancing existing applications not support by the common event framework

Disposition	Guideline
Intention	Support common metric and monitoring infrastructure for those applications that cannot be updated to use the common event framework
Motivation	<ul style="list-style-type: none">▪ Move toward an environment where systems are architected "management-ready" for out-of-the-box integration with Prudential's infrastructure▪ Minimize the amount of metric and monitoring effort required by the application teams
Implication	<ul style="list-style-type: none">▪ For a period of time event management information will be stored in multiple locations▪ Reporting and analysis will need to look in both locations for a complete view

Registry Node

The Registry node enables enhanced management, usage, and governance of the services created as part of the application architecture.

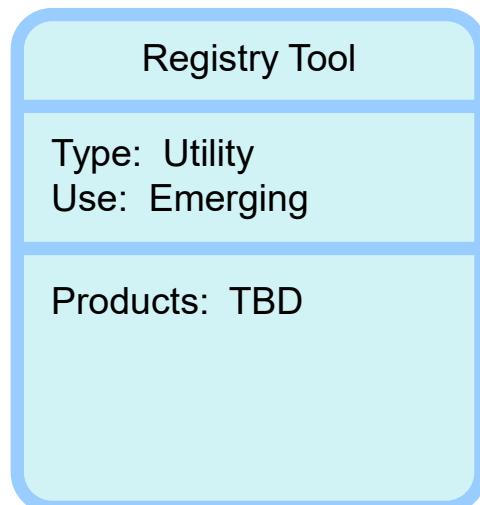


- Application Architecture Level Supported:
 - **Integration - Registry Services**
- Major capabilities: **Service Publishing, Discovery, Dynamic Binding, Tracking of service providers and consumers**
- NFR: **24x7 availability, easily scalable, recoverability**
- Implementation Patterns:
 - **Registry Tool**
 - **Manual Registry**

A registry tool will provide the key capabilities to manage and govern services including service publishing, discovery, dynamic binding and tracking of service providers and consumers.

Status: **Emerging**

Deployment Stack: **Registry Tool**



Services should use a BU-wide service registry

Disposition	Guideline
Intention	All service providers and consumers should register in a common service repository with a standard set of service metadata
Motivation	<ul style="list-style-type: none">▪ Registration will enable better reuse as providers will be able to identify already existing services and eliminate the creation of duplicates▪ Consumers will be easily able to identify services that can be reused
Implication	<ul style="list-style-type: none">▪ Requirements should be identified and a registry tool chosen in the near term▪ Over time as standard registry products are deployed dynamic binding can be used to build more flexible and easily maintained applications▪ Registration will better enable the impact analysis for service changes

A manual registry can be used to track service definitions, and cross reference between service providers and consumers. The manual registry will not enable dynamic binding of services.

Status: **Accommodated**

Deployment Stack: **Manual Registry**



A non-tool based approach to service registry should be used until a tool is selected

Disposition	Guideline
Intention	Manually track service consumers, providers and service location
Motivation	<ul style="list-style-type: none">▪ Service development and deployment will continue regardless of the existence of a registry▪ Management of services must continue to enable better reuse and eliminate creation of duplicate services
Implication	<ul style="list-style-type: none">▪ Once a tool is implemented existing services should be added to the registry▪ Application changes may be necessary to use the registry for binding to services if services bindings are stored within the application

Security Node

The Security node implements the required authentication and authorization capabilities.

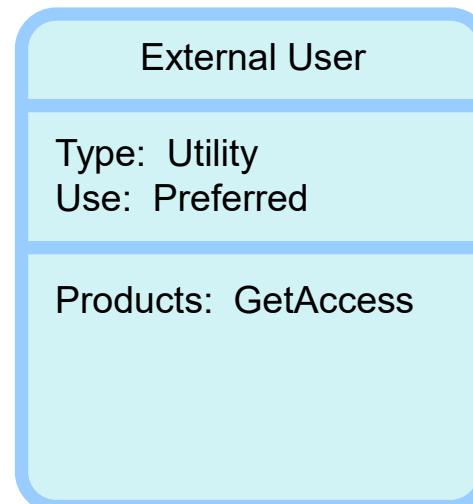
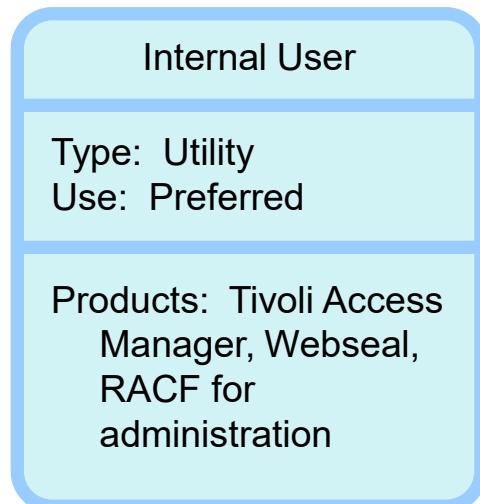


- Application Architecture Level Supported:
 - **Technical Services - Security Services**
- Major capabilities: **Identify and Authentication, Authorization and Access, Control and Entitlement, Logging**
- NFR: **24x7 availability, easily scalable, recoverability**
- Implementation Patterns:
 - **Human Authentication**
 - **System Authentication**
 - **Externalized Authorization**
 - **Application Authorization**

The human authentication pattern enables validating the credentials of a human who is attempting to access a system. There are two preferred deployment stacks for this pattern. The Internal User deployment stack uses Tivoli Access Manager to authenticate Prudential employees, and the External User stack uses GetAccess.

Status: Preferred

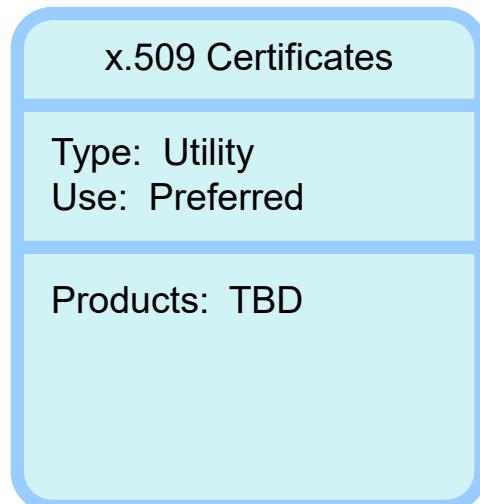
Deployment Stacks: Internal User, External User



The system authentication pattern provides a mechanism to authenticate non-human users of Prudential systems and is based on mutually agreed authentication scheme. This is used for both Internal and External Systems.

Status: Preferred

Deployment Stack: x.509 Certificates



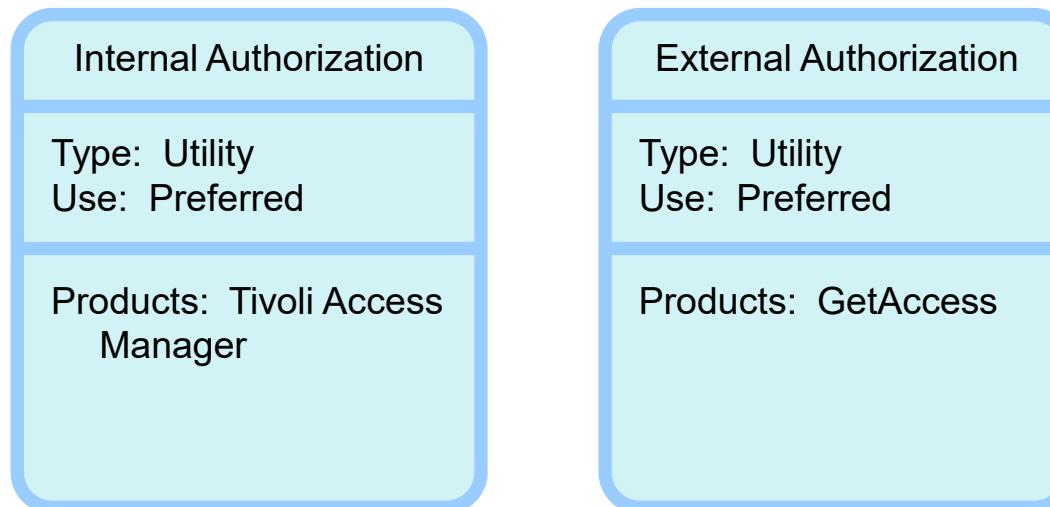
Security for external facing applications will follow CTM standards

Disposition	Policy
Intention	Standard CTM security policies must be followed
Motivation	<ul style="list-style-type: none">▪ Security must be consistent across all applications
Implication	<ul style="list-style-type: none">▪ User access to internal facing applications will use TAM for authentication▪ User access to external facing applications will use Get Access for authentication▪ System access requiring authentication should use x.509 certificates

The externalized authorization pattern leverages the capabilities of an external authorization component to determine if a user should be granted access to a particular resource within the application.

Status: Preferred

Deployment Stack: Internal Authorization, External Authorization



Applications should use externalized authorization when possible

Disposition	Guideline
Intention	Authorization should be implemented and managed external to the application and follow CTM authorization guidelines
Motivation	<ul style="list-style-type: none">▪ Security should be consistent across all applications▪ Reduce the amount of application code needed to implement authorization
Implication	<ul style="list-style-type: none">▪ Developers must design and plan for the role based authorization model defined in Tivoli Access Manager▪ Developers will need to understand the underlying implementation of Tivoli Access Manager authorization▪ Externally exposed applications must consider bridging between GetAccess and Tivoli Access Manager

The application authorization pattern contains functionality within an application for determining if a user should be granted access to a particular resource within the application. No external authorization component or tool is used.

Status: **Discouraged**

Deployment Stack: **Application Authorization**



Applications should use their own authorization capabilities only when they cannot use the capabilities provided by the preferred pattern

Disposition	Guideline
Intention	Limit use of non-preferred patterns to situations where the preferred pattern is not applicable or where existing applications are being enhanced and there is no opportunity to move to the preferred pattern
Motivation	<ul style="list-style-type: none">▪ Security should be consistent across all applications▪ Reduce the amount of application code needed to implement authorization
Implication	

Web Server Node

The Web Server node provides a common access point to web based applications. It hosts static and dynamic content for presentation to consumers.



- Application Architecture Level Supported:
 - **Interaction**
- Major capabilities: **Reverse proxy, Load balancing, Security plug-in, Static / Dynamic content**
- NFR: **24x7 availability, easily scalable**
- Implementation Patterns:
 - **Web Server**

The web server pattern provides a common access point to web based applications, both internally and externally facing. CTM provides three deployment stacks for the web server pattern.

Status: **Preferred**

Deployment Stacks: **SunOne, IBM HTTPS, Microsoft**

SunOne Web Server

Type: Utility
Use: Preferred

Products: SunOne – Internet

IBM HTTPS Web Server

Type: Utility
Use: Preferred

Products: IBM HTTP Server – Intranet

Microsoft Web Server

Type: Utility
Use: Accommodated

Products: Microsoft IIS
.Net environment only

The SunOne should be used for external and IBM HTTP server should be used for internal applications

Disposition	Guideline
Intention	Follow CTM standards for web servers where possible
Motivation	<ul style="list-style-type: none">▪ The web server is integral and tightly coupled to the application platform▪ Use of standard platforms deployed by CTM reduce complexity of application development and deployment
Implication	<ul style="list-style-type: none">▪ The selection between SunOne and IBM HTTP server should follow the CTM guidelines for how to deploy web applications

Interaction Node

The Interaction node provides the user interface to capabilities provided by the application architecture. It enables non-thick client users to access the application functionality and supports portlet, JSP, and rich client capabilities.



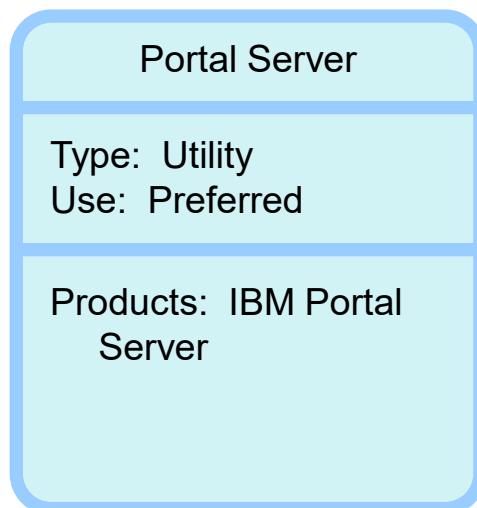
- Application Architecture Level Supported:
 - **Interaction**
- Major capabilities: **User interface presentation, Aggregation of multiple content sources, Personalization, Customization of the user interface, Session Management**
- NFR: **24x7 availability, easily scalable, recoverability**
- Implementation Patterns:
 - **Portal Interaction**
 - **Application Server Interaction**

The portal interaction pattern provides a single point of interaction for dynamic information, applications and processes. Common portal services include:

- Connectivity and integration applications both internal and external
- Personalization of the user interface
- Security features that include an authentication layer to provide controlled access and Single Sign On

Status: Preferred

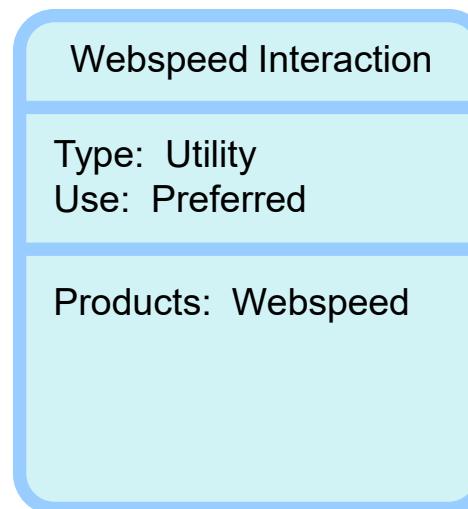
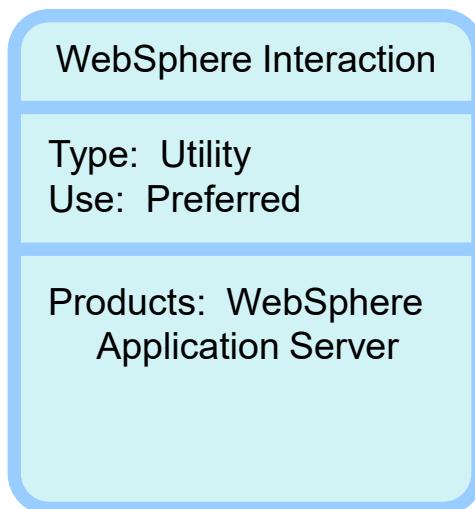
Deployment Stack: Portal Server



The application server interaction pattern enables deployment of servlet, struts, JSP, and other rich client user interfaces. Presentation is specified by the application developer. Personalization is not inherent in the pattern but defined by the developer. Typically a single application screen is presented at one time rather than multiple windows showing multiple applications.

Status: Preferred

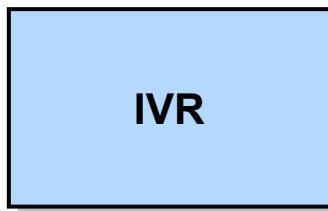
Deployment Stacks: WebSphere Interaction, Webspeed Interaction



The portal interaction pattern should be used for applications unless constrained by application architecture

Disposition	Guideline
Intention	Portal provides a more consistent and flexible pattern for implementing user interfaces
Motivation	<ul style="list-style-type: none">▪ Increased desire to provide more flexible, integrated and personalized interface to application functionality
Implication	<ul style="list-style-type: none">▪ Additional patterns and guide lines must be developed to enable more rich clients (e.g. AJAX) to be deployed within the portal▪ Continue using the application server interaction pattern where it is currently implemented or deemed appropriate by application architecture

The IVR node enables access to the applications through voice devices. IVR, is a phone technology that allows a computer to detect voice and touch tones using a normal phone call. The IVR system can respond with pre-recorded or dynamically generated audio to further direct callers on how to proceed. IVR systems can be used to control almost any function where the interface can be broken down into a series of simple menu choices.

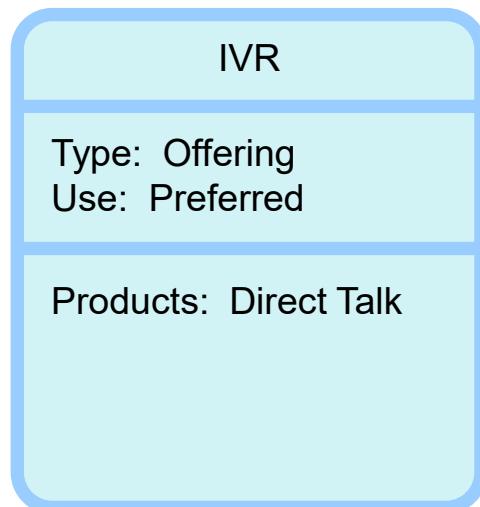


- Application Architecture Level Supported:
 - Interaction
- Major capabilities: **Voice presentation of data for basic service inquiries and responses**
- NFR: **24x7 availability, easily scalable**
- Implementation Patterns:
 - IVR

The IVR pattern enables access to business functionality through voice devices.

Status: Preferred

Deployment Stack: IVR



The IVR pattern should be used to enable applications to interface with phone and voice based interfaces

Disposition	Guideline
Intention	Enable phone and voice based interfaces with applications and application components
Motivation	<ul style="list-style-type: none">▪ Provide a consistent architecture and pattern for interfacing with phone and voice based systems▪ Enable innovative interactions between phone / voice based interfaces and applications
Implication	<ul style="list-style-type: none">▪ Developers will need to start thinking of phone and voice based interfaces as just another interaction medium

Reporting Node

The Reporting node provides a common platform for generating operational reports against all data stores. It includes a common interface for all users accessing reports. A separate tool may be used to support analytic reporting.



- Application Architecture Level Supported:
 - **Information, Component**
- Major capabilities: **Ad hoc reporting, Real time or scheduled reporting, Delivery (electronic, paper, etc.)**
- NFR: **24x7 availability, easily scalable, recoverability**
- Implementation Patterns:
 - **Tool Based Reports**
 - **Native Code Reports**

The tool based reports pattern is a single architecture for viewing, querying, analyzing, and authoring BU reports supporting flexible distribution and multiple output formats.

Status: Preferred

Deployment Stack: Tool Based Reports



The native code reports pattern allows reports to be generated in languages of existing application components (e.g. Progress or Java).

Status: **Accommodated**

Deployment Stack: **Native Code Reports**

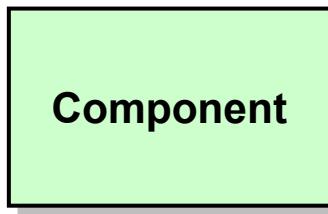


A standard reporting tool should be used for all new report development

Disposition	Guideline
Intention	Move toward a standard easy to use reporting environment
Motivation	<ul style="list-style-type: none">▪ Data and tooling is required to support the critical requirements for Reporting and data analysis
Implication	<ul style="list-style-type: none">▪ Tooling selections need to be finalize and published▪ Training must be provided in the use of new tools

Component Nodes

Component nodes represent the run time environment for each of the components defined in the application architecture (e.g. Party, Product, Contract, Activities, Commissions, etc.). Components can have unique and different deployment stacks.

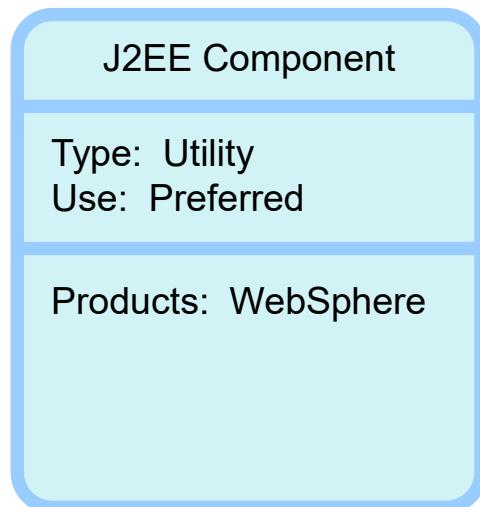


- Application Architecture Level Supported:
 - Component
- Major capabilities: Run time container required by the component, Connection Management
- NFR: 24x7 availability, easily scalable, recoverability
- Implementation Patterns:
 - J2EE Component
 - Progress Component

The J2EE component pattern provides a container that supports the deployment and run time of Java based applications. It provides support for Servlets, EJBs, and other component architectures of the J2EE specification.

Status: Preferred

Deployment Stack: J2EE Component



The Progress component pattern provides a container that supports the deployment and run time of applications developed in the Progress environment. In addition to Progress applications, it also provides an application server that enables exposing Progress capabilities for consumption by applications on the J2EE stack.

Status: Preferred

Deployment Stack: Progress Component



Deploy applications on the stack for which they were built

Disposition	Guideline
Intention	Deploy applications on the stack for which they were built
Motivation	<ul style="list-style-type: none">▪ Each component is build based on policies and guidelines of the application infrastructure and should be deployed upon the stack best suited for the application
Implication	<ul style="list-style-type: none">▪ J2EE components should be designed and developed to run within the standard CTM utility configuration▪ Net is not used for service component development

Business Rules Node

The Business Rules node provides the runtime environment for business rules execution.

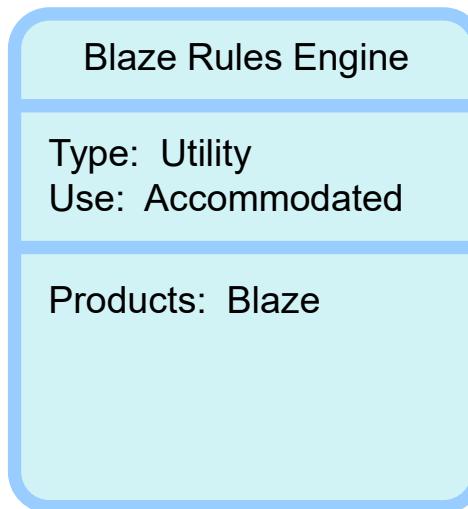


- Application Architecture Level Supported:
 - **Rules**
- Major capabilities: **External rules execution environment, Consumable by any layer of the architecture**
- NFR: **24x7 availability, easily scalable, recoverability, maintainability**
- Implementation Patterns:
 - **External Rules Engine**
 - **Native Code Business Rules**

The external rules engine supports the application architecture policy of externalizing rules from the application. This node provides the runtime environment where the rules are executed.

Status: **Emerging**

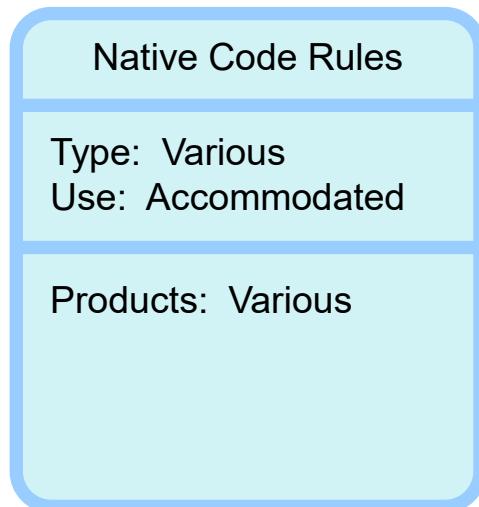
Deployment Stacks: **External Rules Engine, Blaze Rules Engine**



The application architecture guidelines define externalizing rules but not using a rules engine. Rules will be implemented in the environment for which the consuming component is developed (e.g. rule for applications written in Java will be in Java while rules for Progress applications will be developed in Progress).

Status: Preferred

Deployment Stack: Native Code Rules

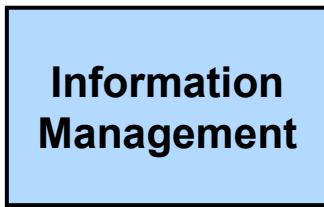


Business rules should be externalized from functional components

Disposition	Guideline
Intention	No rules engine should be deployed until the application architecture is updated to reflect a specific engine
Motivation	<ul style="list-style-type: none">Externalization of rules is an application architecture policy, however, since no engine has been chosen rules will continue to be developed in the same language as the components
Implication	<ul style="list-style-type: none">The rules will need to be developed in native code until a rules engine is selected

Information Management Node

The Information Management node stores and provides access to structured data required for the operation of the business including data related to contracts, products, parties and transactions. It handles data for transaction processing as well as analytics.

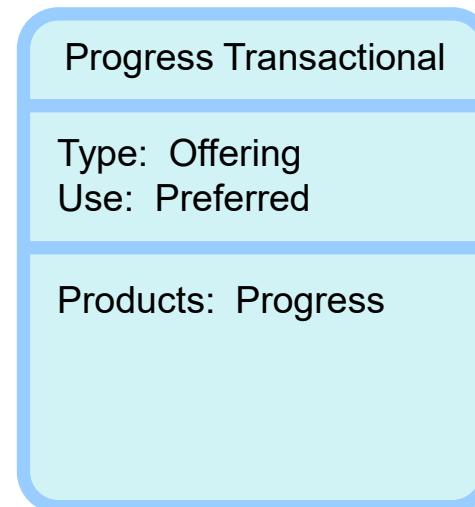
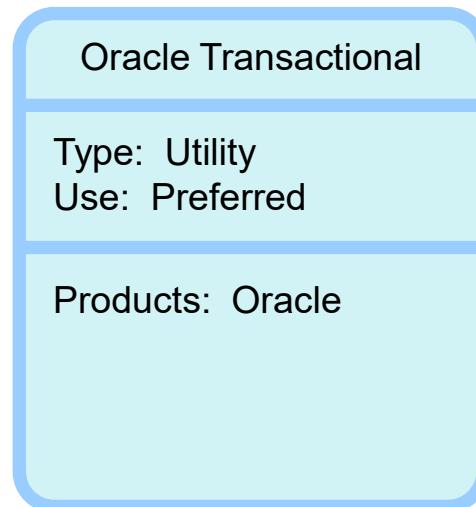


- Application Architecture Level Supported:
 - Data
- Major capabilities: Provides CRUD capabilities for transactional and analytic structured information
- NFR: **24x7 availability, easily scalable, recoverability**
- Implementation Patterns:
 - Transactional Data
 - Analytical Data

The transactional data pattern stores and provides access to structured data for defined business transactions and operations.

Status: Preferred

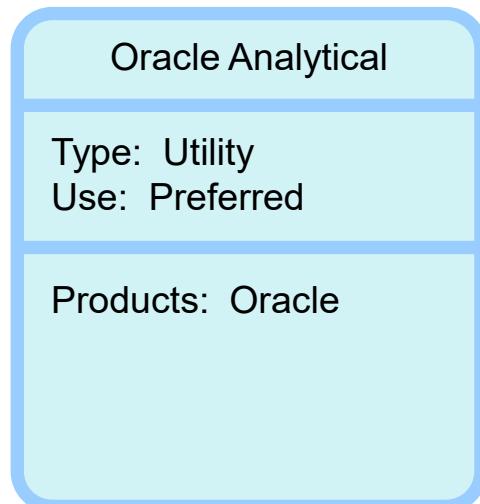
Deployment Stacks: Oracle Transactional, Progress Transactional



The analytical data pattern provides the tooling and run time capabilities necessary to support the unique characteristics of analytical functions and data stores.

Status: Preferred

Deployment Stack: Oracle Analytical



DBMS will be deployed based on Application Architecture needs

Disposition	Guideline
Intention	Each application database has specific requirements for its DBMS, and the appropriate supporting DBMS should be used for each application
Motivation	<ul style="list-style-type: none">▪ Databases must meet the availability requirements of the applications▪ Oracle provides the standard database management system for analytic data stores
Implication	

Content Management Node

The Content Management node serves two main purposes, the creation of content destined for the web server and web application server nodes and the overall management and distribution of that content. The content management node can push/publish content to the web and portal nodes.

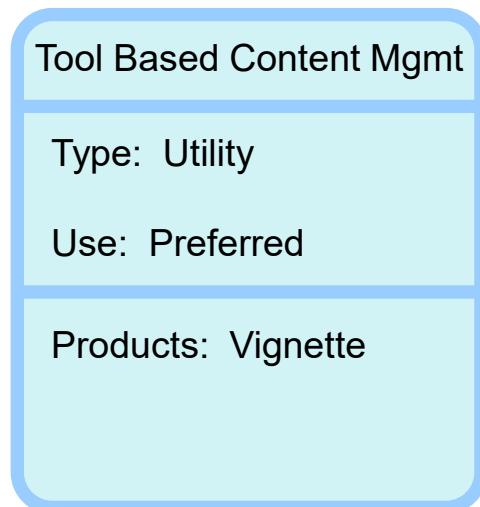


- Application Architecture Level Supported:
 - **Interaction**
- Major capabilities: **Web content management**
- NFR: **24x7 availability, easily scalable, recoverability**
- Implementation Patterns:
 - **Tool Based Content Management**

The tool based content management pattern provides a set of tools and run time for the creation of content destined for the web server and web application server nodes and the overall management and distribution of that content.

Status: Preferred

Deployment Stack: Tool Based Content Management

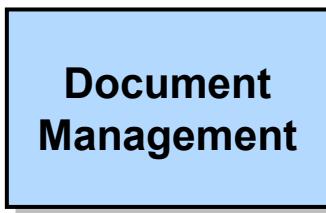


Vignette should be used for managing all content to be published on the web

Disposition	Guideline
Intention	Ensure all BU content is managed through Vignette content manager
Motivation	<ul style="list-style-type: none">▪ Ensure all BU web content is properly and consistently managed▪ Prevent proliferation of content management platform
Implication	<ul style="list-style-type: none">▪ Need to identify all existing platforms and migrate content to Vignette platforms▪ Vignette platform may need to be scaled up and integrated with other systems▪ Some platforms may need to be retired

Document Management Node

The Document Management node stores and provides access to unstructured data required for the operation of the business including images, PDF documents, and voice.

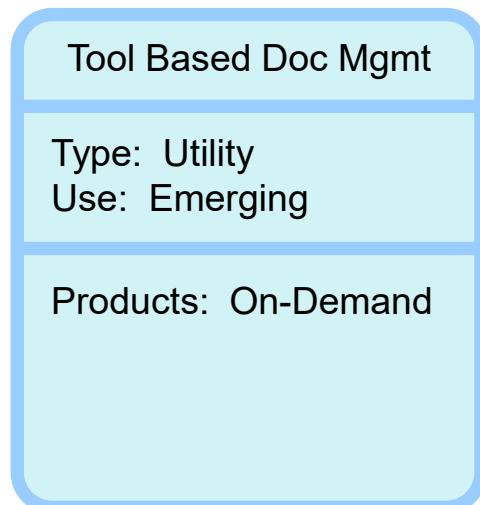


- Application Architecture Level Supported:
 - Data
- Major capabilities: **Provides the repository and CRUD capabilities for unstructured information, Imaging**
- NFR: **24x7 availability, easily scalable, recoverability**
- Implementation Patterns:
 - Tool Based Document Management
 - Native Code Document Management

The tool based document management pattern provides a set of tools and run time for the storage and retrieval of unstructured data.

Status: **Preferred**

Deployment Stack: **Tool Based Document Management**



The Native Code Based Document Management pattern provides document management capability developed in-house in native code.

Status: Preferred

Deployment Stack: Native Code Document Management



Use a document management tool to manage unstructured data

Disposition	Guideline
Intention	Use / reuse specialized tools to perform non business-specific tasks
Motivation	<ul style="list-style-type: none">▪ Move away from building custom solutions where mature vendor solutions exist and remembering we are not a software company▪ Be able to leverage rich document management capabilities of vendor solutions to accelerate solution delivery
Implication	<ul style="list-style-type: none">▪ Standard Document Management tool will need to be identified to avoid proliferation of platforms▪ Migration plan will need to be built for the current custom coded unstructured document handling solutions

Process Node

The Process node provides the container for running and choreographing business processes. Participant interaction with the process layer is enabled through the interaction layer.

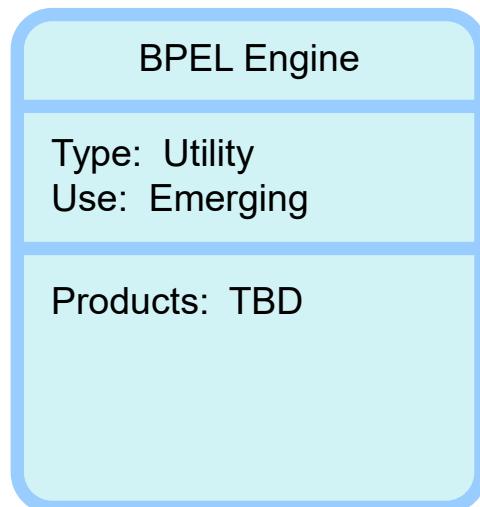


- Application Architecture Level Supported:
 - **Process**
- Major capabilities: **Choreography of services**, **Runtime of composite services**
- NFR: **24x7 availability, easily scalable, recoverability**
- Implementation Patterns:
 - **BPEL Engine**
 - **Native Code Process**
 - **Scheduler**

The BPEL engine pattern uses vendor products to provide a set of tools and a run time container for process orchestration including human tasks and mediation.

Status: **Emerging**

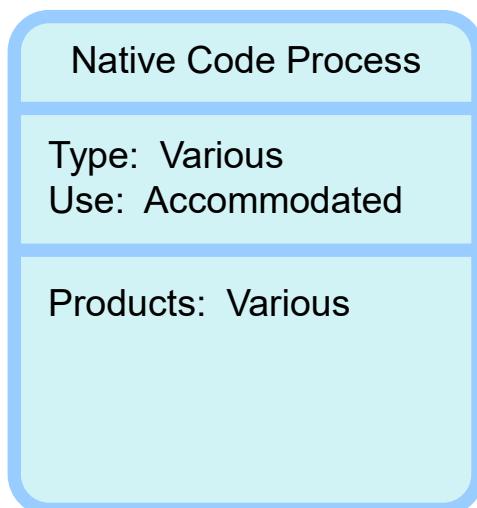
Deployment Stack: **BPEL Engine**



The application architecture guidelines define separating process flow from business function logic. Process logic could be implemented using any reference architecture preferred platform.

Status: **Accommodated**

Deployment Stack: **Native Code Process**



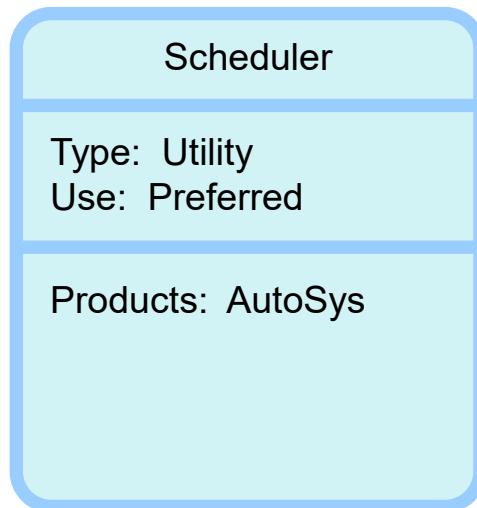
Process choreography should be externalized from the application and use BPEL tool when available

Disposition	Guideline
Intention	Externalize process choreography from the application
Motivation	<ul style="list-style-type: none">▪ Improve flexibility of process choreography to simplify implementation of business process change▪ Use of a process choreography engine will be growing over time
Implication	<ul style="list-style-type: none">▪ Process choreography will have to be developed in native code until a tool is selected▪ A process choreography tool needs to be selected

Scheduler utility is used to choreograph batch oriented processes.

Status: **Preferred**

Deployment Stack: **Scheduler**

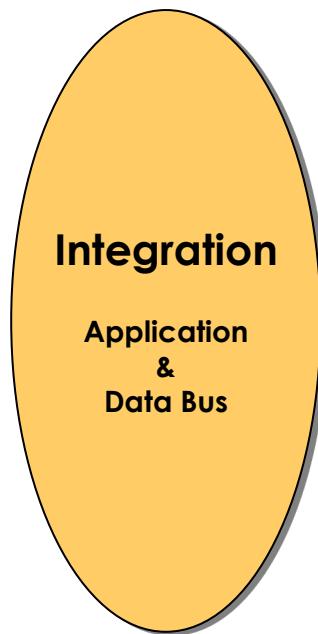


Batch processes will be choreographed by a tool/ script

Disposition	Guideline
Intention	Externalize process choreography from the application
Motivation	<ul style="list-style-type: none">■ Improve flexibility of process choreography to simplify implementation of business process change
Implication	<ul style="list-style-type: none">■ Batch processes will continue to be choreographed by a scheduler

Integration Node

The Integration node provides infrastructure for flexible connectivity and integration of applications and data.

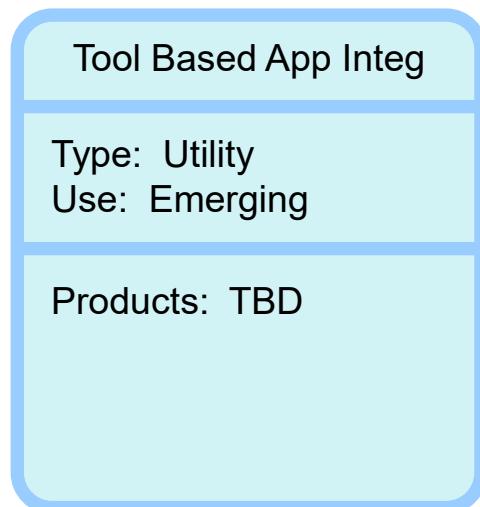


- Application Architecture Level Supported:
 - **Integration**
- Major capabilities: **Routing services, Correlation services, Monitoring services, Transformation services, Extract services, Transport services, Load services**
- NFR: **24x7 availability, easily scalable, recoverability**
- Implementation Patterns:
 - **Tool Based Application Integration**
 - **Native Code Application Integration**
 - **Tool Based Data Integration**
 - **Native Code Data Integration**

The Tool Based Application Integration pattern consists of middleware that unifies and connects services, applications and resources within a business. It is the framework within which the capabilities of a business applications are made available for reuse by other applications throughout the organization and beyond.

Status: Emerging

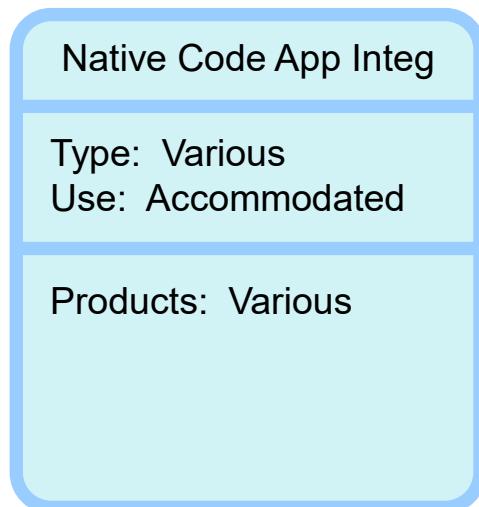
Deployment Stack: Tool Based Application Integration



The native code Application integration pattern uses the component language to develop the required integration capabilities. This approach typically results in a more point to point or more tightly coupled integration effort.

Status: **Accommodated**

Deployment Stack: **Native Code Application Integration**



The Tool Based Application Integration Pattern should be used for new application integrations

Disposition	Guideline
Intention	New applications should begin taking advantage of the Tool based capabilities
Motivation	<ul style="list-style-type: none">▪ Provides a standard, consistent and tool based approach to application integration across the organization
Implication	<ul style="list-style-type: none">▪ The native code integration pattern should only be used until an integration tool is selected and deployed▪ An integration tool needs to be selected ASAP to minimize the amount of point-to-point integration in native code▪ A process should be put in place to define requirements and select an appropriate tool

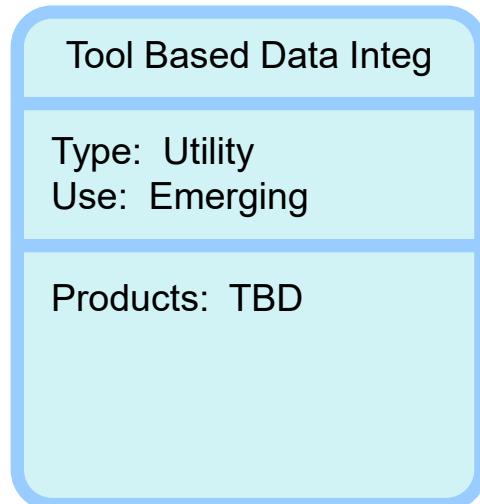
Tool Based Data Integration Pattern

Pattern

The Tool Based Data Integration pattern consists of middleware that unifies , transforms, and transfers information within a business.

Status: Emerging

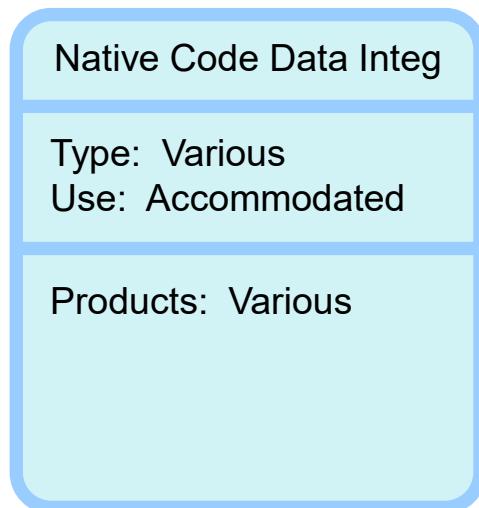
Deployment Stack: Tool Based Data Integration



The native code data integration pattern uses the component language to develop the required integration capabilities. This approach typically results in a more point to point or more tightly coupled integration effort.

Status: **Accommodated**

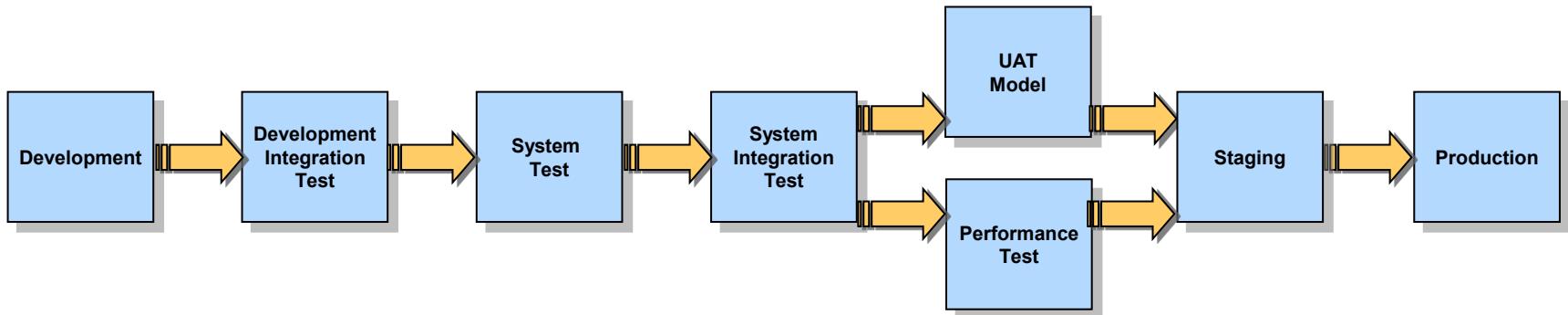
Deployment Stack: **Native Code Data Integration**



The Tool Based Data Integration Pattern should be used for new data integrations

Disposition	Guideline
Intention	New data integration should begin taking advantage of the Tool based capabilities
Motivation	<ul style="list-style-type: none">▪ Provides a standard, consistent and tool based approach to data integration across the organization
Implication	<ul style="list-style-type: none">▪ The native code integration pattern should only be used until an integration tool is selected and deployed▪ An integration tool needs to be selected ASAP to minimize the amount of point-to-point integration in native code▪ A process should be put in place to define requirements and select an appropriate tool

Logical Environments Supporting the Development Lifecycle



Environment Descriptions

Development

- The functional environment where pre-deployment activities are performed. These activities can include coding/compiling, unit-testing individual modules/components, integration and generation of deployable packages. Typically this is a users desktop. Databases may be either local or shared with other developers.

Development Integration

- The functional environment that allows developers to integrate their various modules and perform initial testing to external systems.

System Test

- The functional testing, interface testing, and regression testing will be conducted in this test environment. This is the initial handoff to the QA team.

Systems Integration Testing (End-to-End)

- Systems Integration testing is a test level which verifies the integration of all applications, including interfaces—internal and external—to the organization, with their hardware, software, and infrastructure components in a production-like environment.

Model Office / User Acceptance Test (UAT)

- Customer Acceptance Testing is performed in the Model Office/UAT environment. Customer Acceptance test verifies that the system meets user requirements as specified. It simulates the user environment and emphasizes security, documentation and regression tests and will demonstrate that the system performs as expected to the sponsor and end-user so that they may accept the system.

Performance Test

- This can include load (stress and volume) testing and performance (response time) testing.

Staging

- An environment for testing with production quality data.

Production

- Live use by the business

Environment Characteristics

- The environments provide a structured path from development to production. Progressively more stringent testing and controls are exercised in moving toward the production environment.
- Test environments should provide a complete and controlled ecosystem for the level of testing being performed. It should be possible for one application to call a service provided by another component with repeatable results.
- Typically there is a single environment for each level (dev, system test, production, etc). In instances where multiple projects with differing release dates are developed and tested, it may be necessary to create multiple instances of environments. In most cases, duplicate environments would not be created for system integration test or higher environments.
- Non-production environments do not exist forever. It should be possible to systematically provision and remove environments on a timely basis.
- Each environment should be capable of handling the volume of transactions being tested in that environment. The Performance testing environment should be comparable to production in capability.

Use Preference Definitions

Preferred

- This is the preferred pattern for all future efforts
- All exceptions must be evaluated and approved through the governance process

Accommodated

- Although not the preferred pattern this pattern is accommodated where significant investment already exists and it is too risky to change to the preferred pattern in a single effort
- Accommodated patterns are a limited set of pre-approved exceptions to the preferred pattern
- No new component development should be undertaken with this pattern

Discouraged

- These patterns should not be used except in very specific circumstances
- Special approval must be approved before using this pattern
- Discouraged patterns are allowed if the cost (Tax) of moving to the preferred pattern is 50-75% or more

Emerging

- The pattern will be used with special approval
- Details regarding the implementation are not yet available

Deployment Stack Type Definitions

Base

- Basic operational components and support
- Features and components that provide the base for offerings and utility environments
- Includes switches, routers, firewalls, proxies
- Foundation and wiring

Offering

- A standard platform
- Hardware procured by CTM and funded by the BU
- Configuration parameters are negotiated between CTM and the BU
- Management of the environment is negotiated between CTM and the BU
- Environments may be managed at different levels by different organizations
- A collaborative effort to define, build, manage the environment

Utility

- CTM provided environment
- Fully managed
- Applications conform to a standard structure, deployment model
- Basically a run time container

Agenda

1 RA Vision, Principles, Policies, and Guidelines

2 RA Blueprinting - Business Architecture

3 RA Blueprinting – Application Architecture

4 RA Blueprinting – Information Architecture

5 RA Blueprinting – Technical Infra. Architecture

6 Appendix A: Business Architecture Details



Product - Product Portfolio Planning

❑ Definition

Determine market need for new products and services, and identify internal or external offerings that may address the need. Determine when new products should be launched, and when current products should be modified, suspended, or withdrawn from the market or a channel. Establish strategic direction for the overall portfolio of product and service offerings. Includes developing new product/service strategy needed to achieve corporate growth and financial results targets

❑ Functions and Activities

- Product Market Alignment
- Feature Analysis
- Product Opportunities
- Product Release Strategy

Product - Product Conceptualization

❑ Definition

Creating the concept of a new product. This may be based on formal market or customer need analysis or on unstructured process, based heavily on experience and perception in the market place.

❑ Functions and Activities

➤ Product Conceptualization

Product - Product Definition and Design

❑ Definition

Develop and maintain specific product & service specifications which include benefit, rider, pricing & form rules. This includes testing of risk limits. Create a common enterprise source of all information about products and services, rates, pricing rules, terms & conditions, and risk attributes

❑ Functions and Activities

- Product Requirement Analysis
- Product Design
- Product Scoping

Product - Product Deployment

❑ Definition

Release new products & services for distribution and communicate availability, including any training of intermediaries and employees, and tools required to successfully deploy the product or service.

❑ Functions and Activities

- Product Setup
- Product Training
- Marketing Material Development
- Product Launch

Product - Product Performance and Pricing

□ Definition

Determine and ensure price adequacy by assessing the profitability of proposed product and service offerings; managing market risks, fees, expenses, evaluate and manage exceptions to ensure targeted profitability is met. Measure and manage profitability of products and services offerings against target.

□ Functions and Activities

- Actuarial
- Product Pricing
- Product Performance Analysis
- Product Profitability Analysis

Product - Regulatory Filing

- Definition

Execute & manage filing and regulatory approval process for product rules and forms.

- Functions and Activities

- Filing Preparation
- Filing

Product - Investment Management

❑ Definition

Identifying what underlying assets will be part of the product offering. This includes building relationships with fund managers and identifying investment opportunities and vehicles to ensure that the product meets the stated objectives.

❑ Functions and Activities

- Investment Relationship Management
- New Investment Management Strategy
- Investment Risk Analysis
- Asset Allocation Selection
- Fund Manager Selection
- Alternate Investment Strategy

Sales - Promotion and Brand Management

□ Definition

Understand perception of company's brands and promote the company and its services/products, its intermediary network and company image through brand promotion activities and public relations. Manage the brands so clients automatically associate certain "best of breed" products with the company.

□ Functions and Activities

- Brand and Penetration Analysis
- Brand Awareness
- Brand Promotion

Sales - Sales Generation and Enablement

❑ Definition

Manage the sales lifecycle in order to maximize revenue generation and operational efficiency. Provide leads, tools, training, reports and assistance that can help producers manage their relationships with the company and their customer interactions, improve their “close” rate for new business, and effectively manage their overall book of business.

❑ Functions and Activities

- Sales Generation Ideas
- Sales Education
- Sales Lead Generation
- Illustrations
- Sales Materials

Sales - Licensing and Appointments

❑ Definition

Validating licensing of a firm or financial professional and appointing them for legally selling company's products

❑ Functions and Activities

- Licensing
- Appointment

Sales - Campaign Management

❑ Definition

Develop and execute marketing campaigns based on taking a holistic approach to selling products and improving brand image, including leveraging the company's business intelligence efforts.

❑ Functions and Activities

- Campaign Planning
- Campaign Management
- Campaign Execution

Sales - Sales Compensation

- ❑ Definition

- Create compensation and incentive programs for wholesalers.

- ❑ Functions and Activities

- Wholesaler Comp Strategy and Planning
 - Wholesaler Comp Setup
 - Wholesaler Comp Calculations

Sales - Commissions

❑ Definition

Create compensation and incentive programs for financial professionals. Determine and support commission calculations and commission tiering as well as profit sharing process/agreements.

❑ Functions and Activities

- FP Comp Strategy and Planning
- FP Comp Setup
- FP Comp Calculations
- FP Comp Preference Setup

Sales - Suitability

- Definition

Ensuring that the regulations dictating the suitability of a financial transaction for a customer are adhered to.

- Functions and Activities

- Suitability Review of Transactions

Service - Contract Setup

Definition

Collect, register, and record information related to assets under management. Perform back-office company functions that occur following the submission of the application by the producer. Includes the receiving of the new application, validating producer licensing & appointments, recording contract data, NIGO handling, and policy issuance.

Functions and Activities

- Order Entry
- Exception (NIGO) Handling
- Contract Issuance

Service - Money-In

❑ Definition

Perform activities to collect promised monies. Accept payments in various ways and apply payment to amount due based on pre-defined rules, including management of overdue amounts, write-offs, overpayments and underpayments, sub-pays. Oversee activities of vendors that are responsible for collections.

❑ Functions and Activities

- New Business Collection
- Sub-pay Collection
- Commission Recapture

Service - Contract Non-financial Servicing

□ Definition

Handle the lifecycle of all inbound and outbound contacts across multiple access methods (email, web, fax, mail, etc.), e.g., handling request for non-financial contract change, marketing additional products, riders etc.

□ Functions and Activities

- Contract Administration
- Transfer of Block
- Third Party Administration

Service - Contract Financial Servicing

❑ Definition

All contract service that impacts the financials of the contract. This includes activities like surrender, cancellation, withdrawals etc that generate payment transactions or settlement of monies in that need to be applied to contract.

❑ Functions and Activities

- Valuations
- Contract Surrender
- Payout Processing
- Cancellation
- Outbound Exchange Processing
- Death Benefits
- 72T/ 72U Exchanges
- Partial Withdrawals

Service - Reallocations

❑ Definition

Move money within various investments related with a contract. This may be done automatically (program) by a contract manager or on explicit instructions from the contract holder.

❑ Functions and Activities

- Program Money Movement
- Client Directed Money Movement

Service - Annuitization

❑ Definition

Change the status of a contract from accumulation to payout in response to a maturation event or customer request if allowed by contract rules.

❑ Functions and Activities

- Contract Maturity
- Immediate Annuitization

Service - Payments

□ Definition

Disburse monies by various mechanisms (checks, EFT, wire transfers) to settle financial obligations of the company. This may include payments for loans, surrenders, roll-overs, exchanges, compensation etc.

□ Functions and Activities

- Moneys Out
- Annuity Payments
- Exchange Payments
- Compensation Distribution

Service - Value Added Services

- Definition

Provide any non-contract service to a contract holder or interested party

- Functions and Activities

- Non-Contract Service Planning
- Non-Contract Service Design
- Non-Contract Service Execution

Service - Valuation

- Definition

Get the valuation of a contract by accumulating values of all underlying assets

- Functions and Activities

- Valuation

Service - Correspondence Management

□ Definition

Generate, print, assemble, and issue (mail, digital) communication and policy and product documentation to contractors agents and service providers.

□ Functions and Activities

- Generate Correspondence
- Print Correspondence
- Assemble Correspondence
- Issue Correspondence

Client and Channel Management - Market Research

□ Definition

Gather, store, analyze data, and draw conclusions based on customer needs and review of industry and competitive landscape. Identify trends on buying behaviors, risk profiles, customer segmentation by agent channel, customer segmentation by product, etc.

□ Functions and Activities

- Industry Analysis
- Client Analysis
- Competitive Analysis
- Existing and Trends
- Client Behavior Modeling
- Market Maturity/ Readiness Analysis

Client and Channel Management - Market Segmentation

□ Definition

Classification of market based on market analysis in order to help focus on and better serve the needs of clients at various levels, including: Contractholder, Financial Professional (FP), and/or Firm

□ Functions and Activities

- Contractholder Segmentation
- Firm Segmentation
- FP Segmentation

Client and Channel Management - Channel Management

❑ Definition

Establish the strategy and accompanying plans on how the annuity company interacts with distribution partners (e.g., wholesalers, agents) in order to attract and maintain sales for the company's products and services. This includes partitioning distribution mechanisms, creating producer incentive compensation programs that are customized for each type of distribution channel.

❑ Functions and Activities

- Channel Strategy
- Channel Segmentation and Planning
- Acquisition of New Firms
- Firm Onboarding

Client and Channel Management – Contract Holder Relationship Mgmt

❑ Definition

Develop and execute the strategy for maintaining and improving the ongoing relationships with the contractholders. This also includes monitoring and collecting metrics.

❑ Functions and Activities

- Contractholder Relationship Strategy
- Contractholder Relationship Management
- Contractholder Satisfaction Management
- Contractholder Activity Management

Client and Channel Management - Firm Relationship Management

❑ Definition

Develop and execute the strategy for maintaining and improving the ongoing relationships with the firms selling the company's products. This also includes monitoring performance and collecting metrics to help the partners sell more of company's products.

❑ Functions and Activities

- Firm Relationship Management
- Firm Activity Management
- Firm Satisfaction Management
- Firm Effectiveness, Performance & Profitability
- Product Release Planning by Firm

Client and Channel Management - FP Relationship Management

❑ Definition

Develop and execute the strategy for maintaining and improving the ongoing relationships with the financial professionals (FP's) selling the company's products. This also includes monitoring performance and collecting metrics to help the FP's sell more of company's products.

❑ Functions and Activities

- FP Relationship Management
- FP Activity Management
- FP Satisfaction Management
- FP Effectiveness, Performance & Profitability

Client and Channel Management - Client Profile

❑ Definition

Maintaining the client profile including all information that may help the company to improve its service and relationship with the client. The information may be collected from the annuity application and during the course of the company's ongoing relationship with the client. This is effectively the source of information regarding the client.

❑ Functions and Activities

- Contractholder Profile Setup
- Contractholder Profile Maintenance
- Firm Profile Setup and Maintenance
- FP Profile Setup and Maintenance

Client and Channel Management - Asset Retention

❑ Definition

Define vision, goal, and strategy for ensuring continued growth of net AUM through maintaining existing contracts. This also includes working with sales and service organizations to ensure execution of the strategies.

❑ Functions and Activities

- Asset Retention Strategy & Planning
- Execution

Risk and Financial Management- Risk Management

- Definition

Develop guidelines and plans for managing environmental, operational and other types of risks (but not including financial risks)

- Functions and Activities

- Environmental Risk Management
- Operational Risk Management
- Overall Risk Management

Risk and Financial Management- Regulatory and Compliance

□ Definition

Define vision, goals, corporate guidelines, governance and approach for managing exposure to risk that is acquired. Ensure that the applicable rules and regulations established by regulatory bodies are complied with.

□ Functions and Activities

- Regulatory Monitoring
- Product Regulatory Analysis
- Regulatory Compliance (AML, OFAC)
- Proactive Regulatory Sensing

Risk and Financial Management- Legal

Definition

Understand and interpret various laws, legal precedents and regulations applicable to the business including design of products and litigations of all types

Functions and Activities

- Legal Interpretation
- Legal Environment Monitoring
- Litigation Management

Risk and Financial Management- Financial Risk Management

❑ Definition

Understand, model, and manage risks related to market positions of the investments underlying the products. Also includes actuarial analysis regarding hedging and reserving.

❑ Functions and Activities

- Market Risk Management
- Actuarial Financial Risk Analysis
- Reserving

Risk and Financial Management- Financial Management

□ Definition

Establish guidelines and policies for Corporate financial management, including credit policy, captives policy, AP/AR policy, regulatory reporting and compliance, to help ensure the ongoing financial soundness of the company. It also includes any decisions regarding the investment of company's capital.

□ Functions and Activities

- Financial Policy
- Capital Investment Management
- Treasury
- Forecasting
- Fund Trades

Risk and Financial Management – Financial Reporting and Metrics

❑ Definition

Create and provide financial & metrics reports and views (e.g. ad hoc and preformatted reports; summary and drill-down reports) for internal, external, as well as regulatory purposes.

❑ Functions and Activities

- Statutory Reporting
- Internal Financial Reporting
- External Non-Regulatory Reporting

Risk and Financial Management – Merger and Acquisition

□ Definition

Develop strategy of growing the company by merger with an external entity. This includes developing business goals, identifying targets, managing bids and proposals as well as executing once the merger/ acquisition is approved by regulators and relevant stakeholders.

□ Functions and Activities

- M&A Strategy and Planning
- M&A Opportunity Identification
- M&A Bids and Proposals
- M&A Execution
- Acquire and Transfer of Block of Business
- Carrier Acquisition

Risk and Financial Management – Business Continuity

□ Definition

Planning response to catastrophic business disruption events like pandemics, floods, power outages etc. Includes all aspects needed to ensure a timely resumption of business operations. This includes coordination activities but not the actual execution of the disaster recovery plans, which may be carried out anywhere in the organization.

□ Functions and Activities

- Business Continuity Planning
- Business Continuity Coordination
- Business Continuity Testing
- Disaster Recovery

Business Mgmt and Support – Communications and Public Relations

❑ Definition

Communicate company position and actions during regular course of business, major events, crises, etc. through multiple media. Plan, manage, execute and communicate results of corporate citizenship and philanthropy programs. Address regulatory complaints.

❑ Functions and Activities

- Communications Planning
- Communications Management
- Perception Analysis
- Crisis Management
- Internal Communications

Business Management and Support – Security and Privacy

□ Definition

Develop and manage the security procedures necessary to protect the physical facilities, the intellectual capital that represents the company's confidential and proprietary data, as well as privacy of client information

□ Functions and Activities

- Physical Security Management
- Information Security Management
- Intellectual Property Management
- Privacy Management

Business Management and Support – Human Resources

❑ Definition

Manage the insurance company's human resources activities: e.g., employee classification, recruitment, compensation, employee relations, and training.

❑ Functions and Activities

- Staffing
- Personnel Retention
- Personnel Compensation
- Employee Relations
- Employee Benefits

Business Management and Support – Training and Knowledge Mgmt

❑ Definition

Provide tools, services, and learning opportunities to teach and develop employees and business partners.

❑ Functions and Activities

- Knowledge Need Analysis
- Training Development
- Training
- Knowledge Harvesting
- Channel Education Support

Business Management and Support - Accounting

Definition

Perform accounting functions according to prescribed accounting procedures for the industry. Manage accounts receivable and accounts payable to ensure negotiated / contracted terms are met. Manage general ledger accounts. Create and disseminate internal, external and statutory reports of profit and loss (e.g. 10-K, Annual Report). Prepare and file tax reports. This also includes the budgeting activities.

Functions and Activities

- GL
- AP
- AR
- Budgeting
- Tax
- Cash and Payment Reconciliation

Business Management and Support - Auditing

❑ Definition

Conduct audits of General Controls, Financial Controls and IT Controls and report results to the firm's Audit Committee. This also includes reporting on security, privacy, and SOX.

❑ Functions and Activities

- Financial Auditing
- Operational Auditing
- Security and Privacy Auditing
- SOX Financial Auditing

Business Management and Support – Information Technology

□ Definition

Manage the Information Technology resources, including software, hardware and people, to effectively support the business processes and business direction.

□ Functions And Activities

➤ Plan

- Project Portfolio Planning
- IT Budgeting and Forecasting
- Reference Architecture

- Program Management
- Project Management

➤ Build

- Solution Architecture
- Solution Design

- Solution Construction
- Solution Deploy

➤ Operate

- Problem Management
- Infrastructure Support

- Production Support
- Information Management

➤ Assure

- Software Quality Assurance
- Metrics

- Financial Monitoring
- Governance

Business Management and Support – Analytics and Reporting

❑ Definition

All non-financial reporting needed to monitor and improve company's execution of business.

❑ Functions and Activities

- Project Portfolio Reporting
- Performance Reporting

Business Management and Support – Procurement and Vendor Mgmt

❑ Definition

Establish guidelines to evaluate vendors based on requirements. Negotiate contracts. Once engaged, integrate vendors into day-to-day operations and manage ongoing relationship. Monitor overall vendor performance against contract.

❑ Functions and Activities

- Vendor Relationship Management
- Procurement

Business Management and Support – Document and Content Management

□ Definition

Manage the documents, contents, images, audio, video, and other unstructured content related to the business and make them available to stakeholders as dictated by rules and policies.

□ Functions and Activities

- Content Management
- Document Management
- Image Services
- Print Services

Business Management and Support – Facilities Management

❑ Definition

Handle all the acquisition and operations of the physical facilities such as hours of operation, maintenance and repair of the facilities HVAC Units, Office equipment, etc.

❑ Functions and Activities

- Facilities Acquisition
- Facilities Maintenance