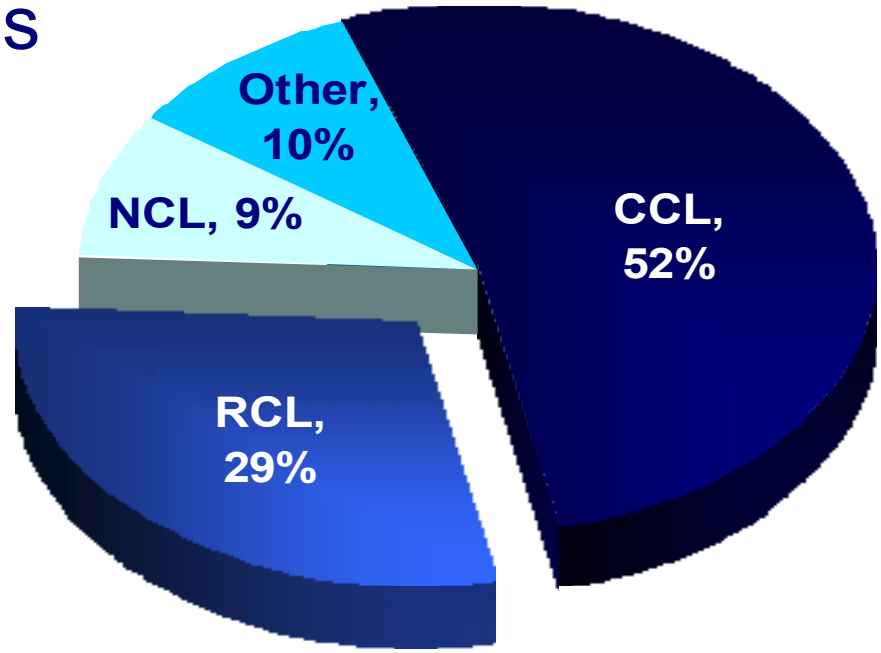
SOA implementation - Lessons Learnt.

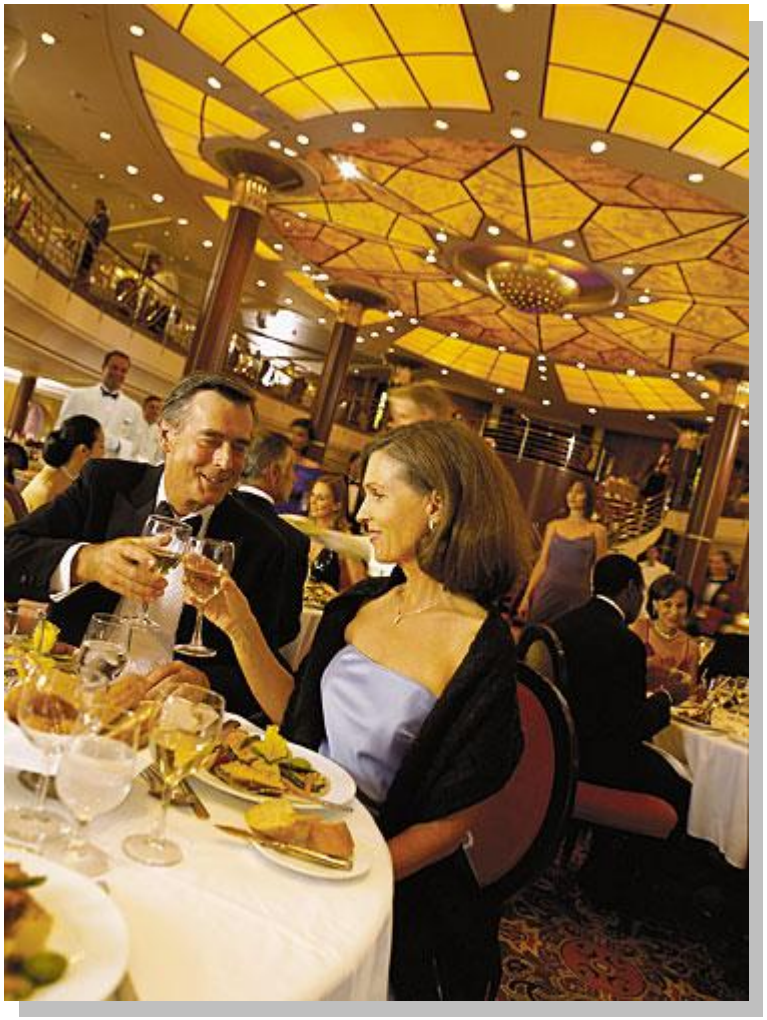
Prashant Naik
IT Chief Engineer
pnaik@rccl.com

Royal Caribbean Cruises & Celebrity Cruises

- Second-largest cruise company
- 29 ships – 60,500 berths
- Strong brands
  - **Royal Caribbean International**
  - **Celebrity Cruises**



Celebrity – The Premium Brand



- **10 ships, approximately 16,500 berths**
- **4 Millennium-class ships: 91,000 tons, state-of-the-art**
- **Superior service**
- **Most environmentally-friendly ships in the world**

Royal – A Vacation Adventure

- **19 ships**
  - **approximately 44,000 berths**
- **5 Voyager-class ships**
- **Young, innovative fleet**
- **Average customer age: 42**
- **Most preferred brand ***
- **Most versatile brand**
- **Most global brand**

Project Churchill's long-term objective is to implement an industry leading reservations platform for RCCL to become a vacation company

Short-term objectives

- ✓ Project Churchill will **transform** the way RCCL does business, paving the way for it to become a vacation company
- ✓ Project Churchill's reservation system will drive efficiency, increasing revenue per customer, and improving RCCL's bottom line

Go live Date September 1, 2007

- We will focus on delivering the following benefits
  - Future Systems **flexibility** through business rules implementation
  - More **flexible** packaging for non-cruise business
  - Improved Revenue Management capabilities
  - **Integration** of Reservations and CRM
  - Internationalization

Pain points and corresponding benefits

Pain Points	Benefits/Capabilities
1. Revenue Management Systems are not **integrated** with Transactional Systems	✓ Provide rule-based pricing structure that is tied to transactions ✓ Provide rule-based logics that can be changed and implemented **instantly** to respond to market shifts and demand ✓ Provide stateroom level probability

Project Churchill's long-term objective is to implement an industry leading reservations platform for RCCL to become a vacation company

Pain points and corresponding benefits

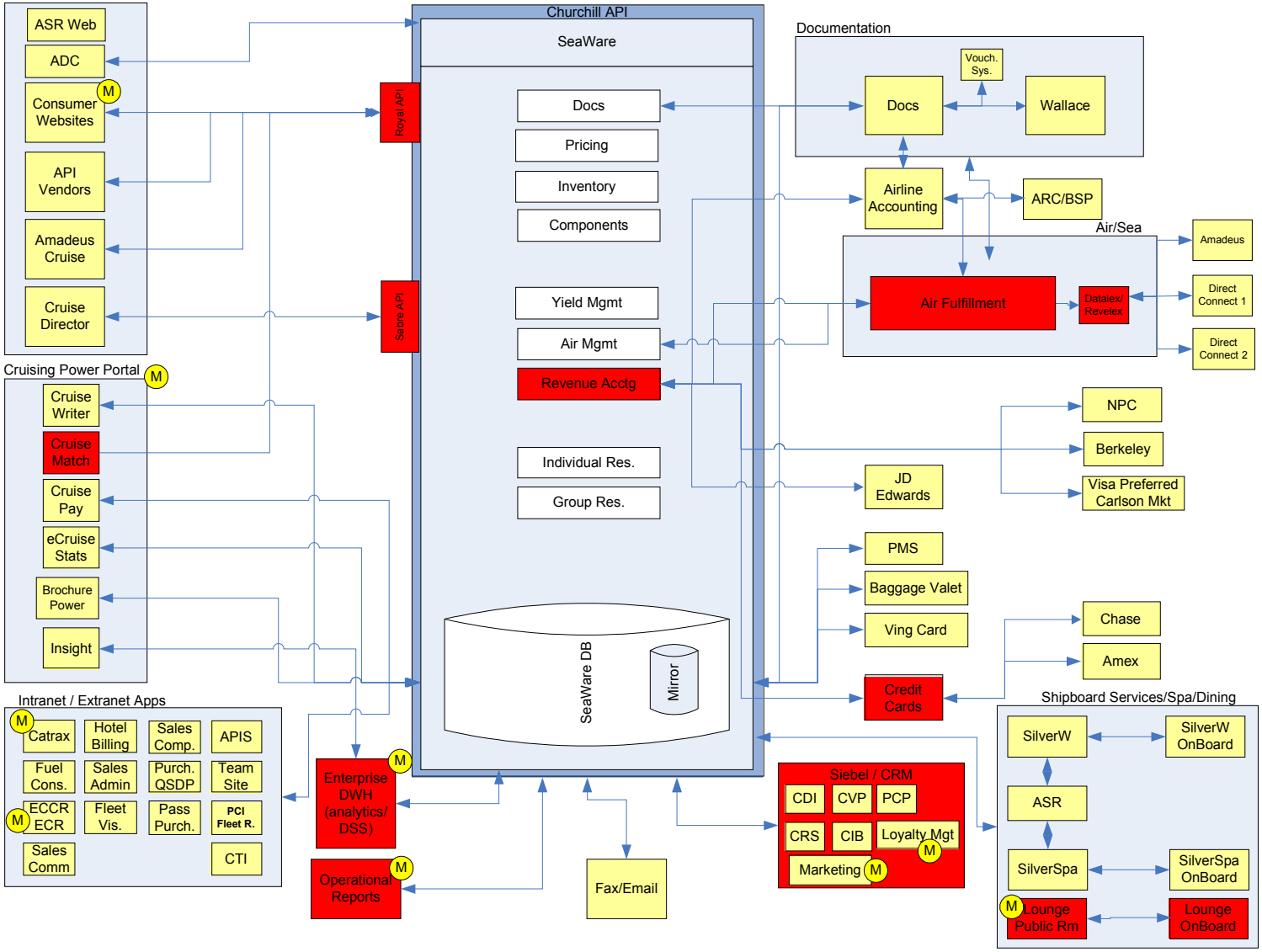
Pain Points	Benefits/Capabilities
1. Inability to address business complexity and business rules to **respond to marketing demand**	✓ Provide **rule-based business policies for business transactions**, e.g., insurance, taxes and fees, payment schedule, etc. ✓ Provide **flexible** rule-based decision making engine for pricing, cost structure, discounts, and promotions ✓ Provide **flexible** rule-based engine for inventory management with yield maximization
1. Inability to create discount allocations and preserve inventory for higher-yielding guests	✓ Utilize business rules on **guest segmentation** to offer discounts or command price premiums ✓ Combine stateroom categories to maximize stateroom price and sales
1. Inability to support **customized offers and improved market segmentation**	✓ Analyze customer profiles for **CRM** and marketing purposes; suggest products and services based on a client profile without relying on a reservation agent (CRM); view service history for a particular client ✓ Target message to specific markets to promote market applicable products and services

Project Churchill's long-term objective is to implement an industry leading reservations platform for RCCL to become a vacation company

Pain points and corresponding benefits

Pain Points	Benefits/Capabilities
1. **Move beyond the standard** cruise as the standard package	✓ Provide **dynamic packaging** capability, i.e., create **flexible** vacation packages, mix and match packages and a-la-carte products in one booking, sell additional hotel nights on demand ✓ Create multiple bookings at one time; include or exclude an FIT booking into or from a group ✓ Provide capability to sell back-to-back and segmented cruises
1. Internationalize the reservation system	✓ Build country and office **specific business rules** to support differences in pricing, commissions, eligibility, and option length
1. Enhance commission flexibility	✓ Provide **flexible** rule-based commissions; associate specific commissions and, payment schedules to each travel agency ✓ Track travel agents' performance and provide a **performance-based commission** structure

Project Churchill system scope



Project Churchill Functional Layer View

Churchill Architecture Model - Enterprise Functional Layer View

User

Employee | RCCL Operation Admin | Consumer | SDC Travel Partner External tools | CMO TA's | Revenue Mgt | Doc | Accounting | RCCL - CC Reservation Agent | Air | RCCL Product Staffs

Presentation Layer

Existing Existing Existing Appls UI's | RCI.COM UI | CEL.COM UI | Sabre Cruise Director GDS UI | Amadeus GDS UI | External Partner Tools - UI | SeaWeb or CruiseMatch Converged UI | Revenue Mgt UI | Documentation UI | Accounting UI | SeaWare Citrix based THICK Client | Airware Citrix based THICK Client

Business Services Layer

Royal API Mapper

Churchill Service API

Non-Core Services: Doc. | Guest Acct | ASR | Others TBD

Core Reservation Services: Cruise | Air/Sea | Hotel | Add-on | Bus,Rail | Payment | Sp Req

Integration Layer

Service Registry

ESB: Routing | Mediation

ASR Adapter | Doc. Adapter | Guest Acct Adapter | Seaware Adapter | AirWare Adapter

Business Component Layer

Integrated Existing Applications: CruisePay | CATRAX | TTG | SilverWare | SilverSpa | Encore | ADC | ASR

Enhanced & Integrated Existing Applications: Doc. (Voucher) | Guest Acct

Seaware System: BizLogic Server | Service Broker | Event Processor | Inventory Servers | Ship | Hotel | Air | Product | Reservation | Selling Limits

AirWare GDS

Non-Churchill Appl.

CRM | JDE | EMP | SCM

Common Application Services Layer

Logging | Event Management | Common Services

Data Access Layer

Existing DBs | EDW + Report | Churchill DB | Seaware DB | Air DB

Infrastructure Services Layer

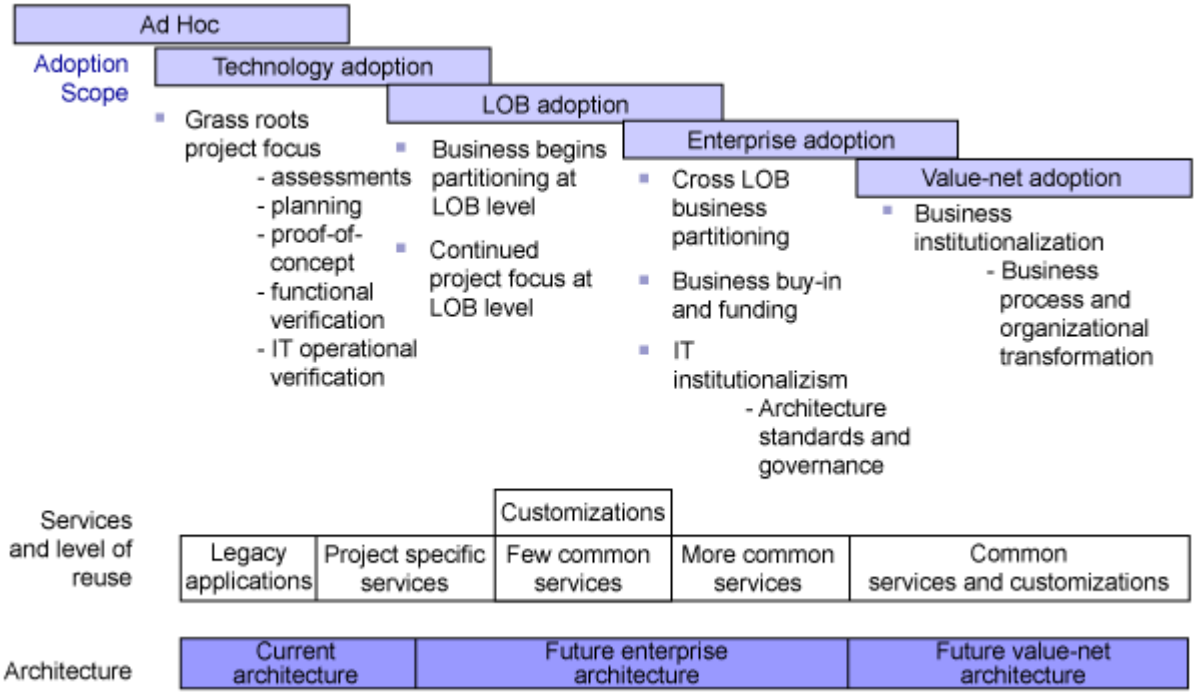
Security Management | System Management

Notes: Appl (NO Func. Enhancement) | Enhanced Functions | New /re-write Functions | 3rd party Package

Lesson #1 : SOA Adoption is a Gradual process.

Scope of Adoption:

- Ad hoc usage evolves into **Technology adoption: Tools, technologies, standards, methods, and patterns that enable creation of proofs of concept**.
- **LOB adoption: Sharing services across LOB eliminates redundancy & increases flexibility.** Leads to consolidation of business functions across lines of business.
- **Enterprise adoption**: **Standards, governance, and institutionalization of IT occur** - - protocols, tools, standards, service models, funding models.
- **Value-net adoption:** Extension of services across organizational boundaries.
- **Reuse**: From wrapping of legacy, to project specific services to common enterprise- wide services.
- **Architecture**: From Project specific to more strategic services-oriented enterprise architecture.



IBM Developerworks

IBM's Seven Stages of Service Integration Maturity Model (SIMM)

Lvl 7 — **Dynamically reconfigurable services (eco-system integration)**
The organization now has a dynamically re-configurable software architecture. It can **compose services at run-time using externalized policy descriptions, management, and monitoring.**

Lvl 6 — **Virtualized services (virtual infrastructure)**
The organization now creates a virtualized infrastructure to run applications. It achieves this level after **decoupling the application, its services, components, and flows.** Now the infrastructure is more finely tuned, and the **notions of the grid** and the grid service render it more **agile**.

Lvl 5 — **Composite services (supply-chain integration)**
Now the organization extends its influence into the value chain and into the **service eco-system**. Services form a **contract among suppliers, consumers, and brokers** who can build their own eco-system for on-demand interaction.

Lvl 4 — **Simple services (process integration)**
The organization embarks on the **early phases of SOA** by defining and **exposing services for consumption internally or externally** for business partners -- not quite on a large scale -- but it acts as a service provider, nonetheless.

Lvl 3 — **Componentized (functional integration)**
At this level, the organization **componentizes and modularizes** major or critical parts of its application portfolio. It uses legacy transformation and renovation methods to re-factor legacy systems with clear component boundaries and scope, exposing functionality in a more modular fashion. The **integration between components** is through their **interfaces and the contracts** between them.

Lvl 2 — **Integrated (application integration)**
The organization moves toward some form of **EAI** (Enterprise Application Integration), albeit with **proprietary connections** and integration points. The approaches it uses are **tailored to use legacy systems** and attempt to dissect and re-factor through data integration.

Lvl 1 — **Silo (data integration)**
The organization starts from **proprietary and quite ad-hoc integration**, rendering the architecture **brittle** in the face of change.

Lesson #2 : Focus on how to do SOA right instead of getting lost in the Alphabet Soup.

ESB

SOA

SOA Infrastructure

Abstraction

SOA

Fabric

EAI

Service Network

EDA

Service Grid

- There are many styles of SOA implementation.
- Focus on your goals: Reuse, Governance, Reduced Integration Cost, Agility.
- It really doesn't really matter what the precise definition of SOA is, as long as you implement SOA in a way that addresses the business problems you set out to solve.

Lesson #3 : Do Not approach Service Oriented Architecture like Traditional distributed computing.

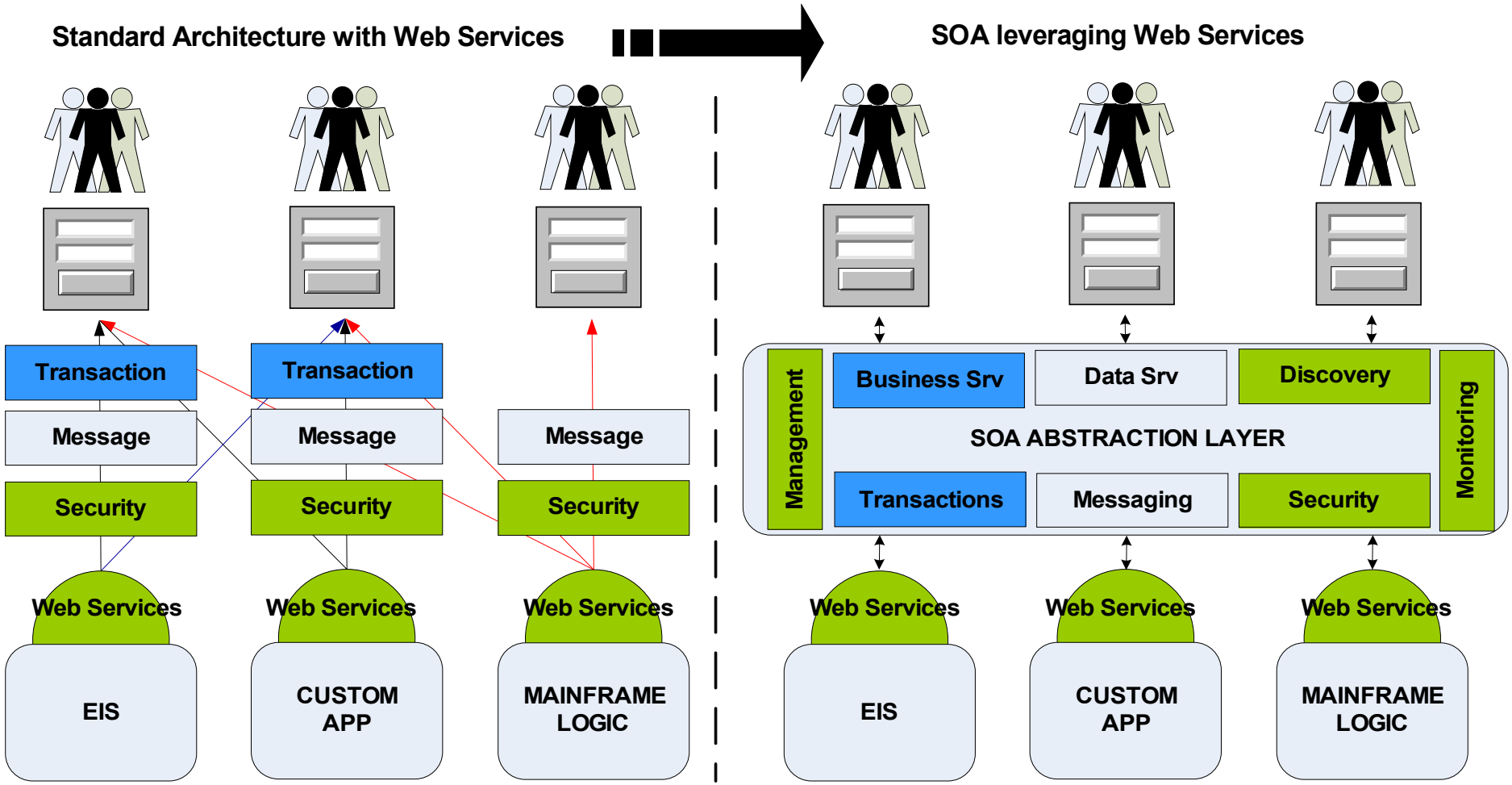
Traditional Distributed Computing	Service Oriented Architecture
➢ **Designed to Last**.	➢ **Designed to change**.
➢ **Tightly Coupled**.	➢ **Loosely Coupled, Agile and adaptive.**
➢ **Integrate Silos.**	➢ **Compose Services.**
➢ Code Oriented.	➢ Process Oriented.
➢ **Long Development cycle.**	➢ **Interactive and iterative development cycle.**
➢ Cost centered.	➢ Business centered.
➢ Middleware makes it work.	➢ Architecture makes it work.
➢ Favors Homogenous Technology.	➢ Favors Heterogeneous Technology.
➢ **Focus on performance during coding.**	➢ **Focus on performance during design.**
➢ **Fast initial Time-To-Market.**	➢ **Fast Time-To-Enhancement.**

Lesson #4 : SOA is not the only choice for EVERY project.

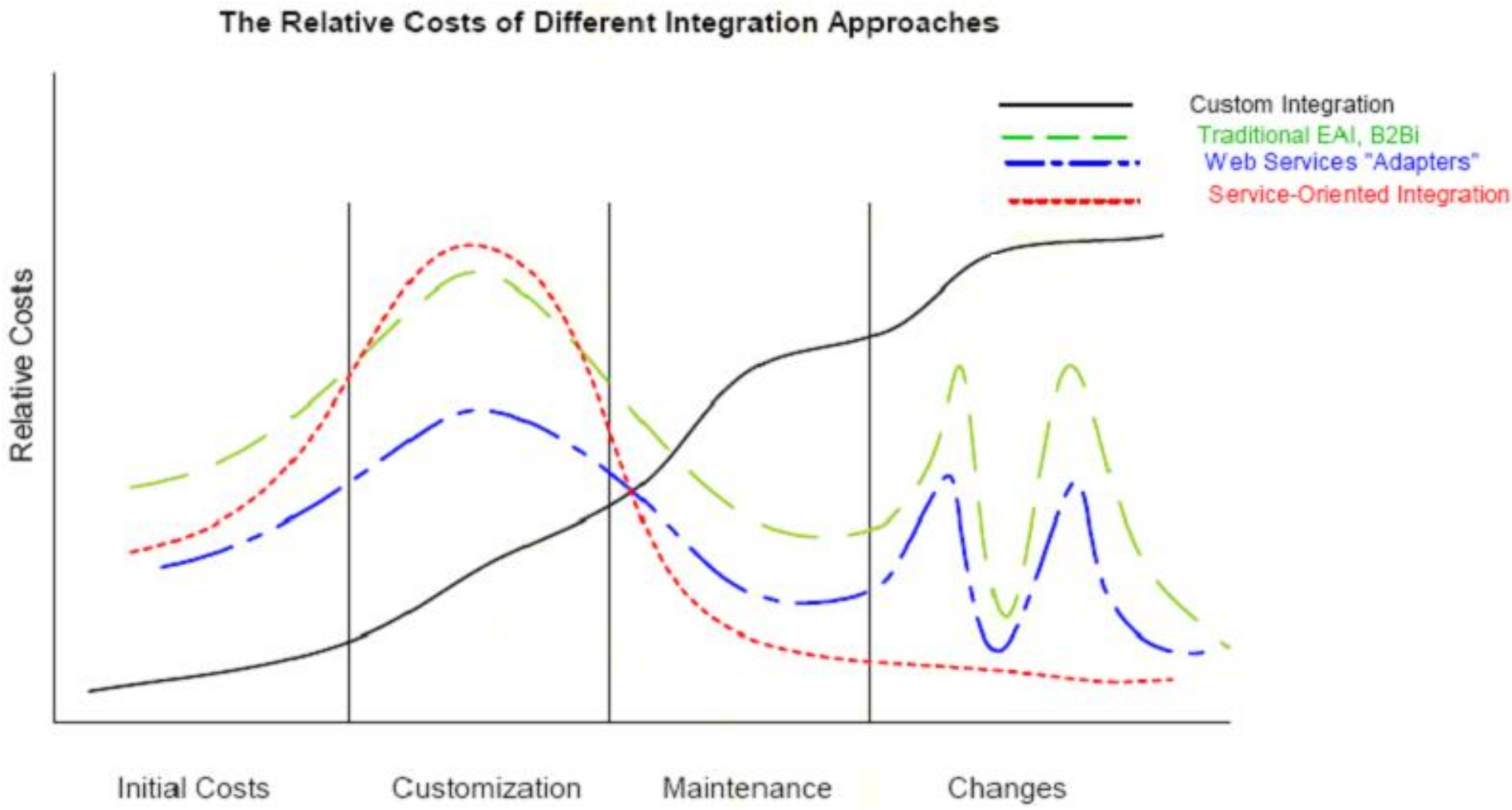
IF the only tool you have is a hammer, you will see every problem as a nail.

- **What is the anticipated lifespan of the application ?**
  - Since, ROI for SOA comes as an application evolves over time, SOA is not a good fit for short lifespan applications.
- **How critical is the application to your business ?**
  - If the target application is intended to enhance your business in a way that directly increases your ability to serve your customer, overcome performance barriers, or break new and innovative ground in accomplishing your company's primary business objectives, then, the target application is critical to your business.
- **How much change do you anticipate during the lifespan of this application ?**
  - If the application in question is, for the most part, isolated from many of the other systems in your business, or if, over the life of the application, the requirements and demands of the application are relatively stagnant, SOA may not be the right approach for the project.
- **Do you anticipate that the technical requirements for this application will change over time?**
  - SOA is exceptionally valuable when building applications that may undergo significant change in: Platform requirements, Load requirements, Integration requirements, access control requirements, scalability, business dynamics.
- **Does your application have an end-user component or is it purely a back end application?**
  - SOA can aide in both easing the pain of frequently changing business flow or logic in applications, while leveraging existing systems and core applications where appropriate.

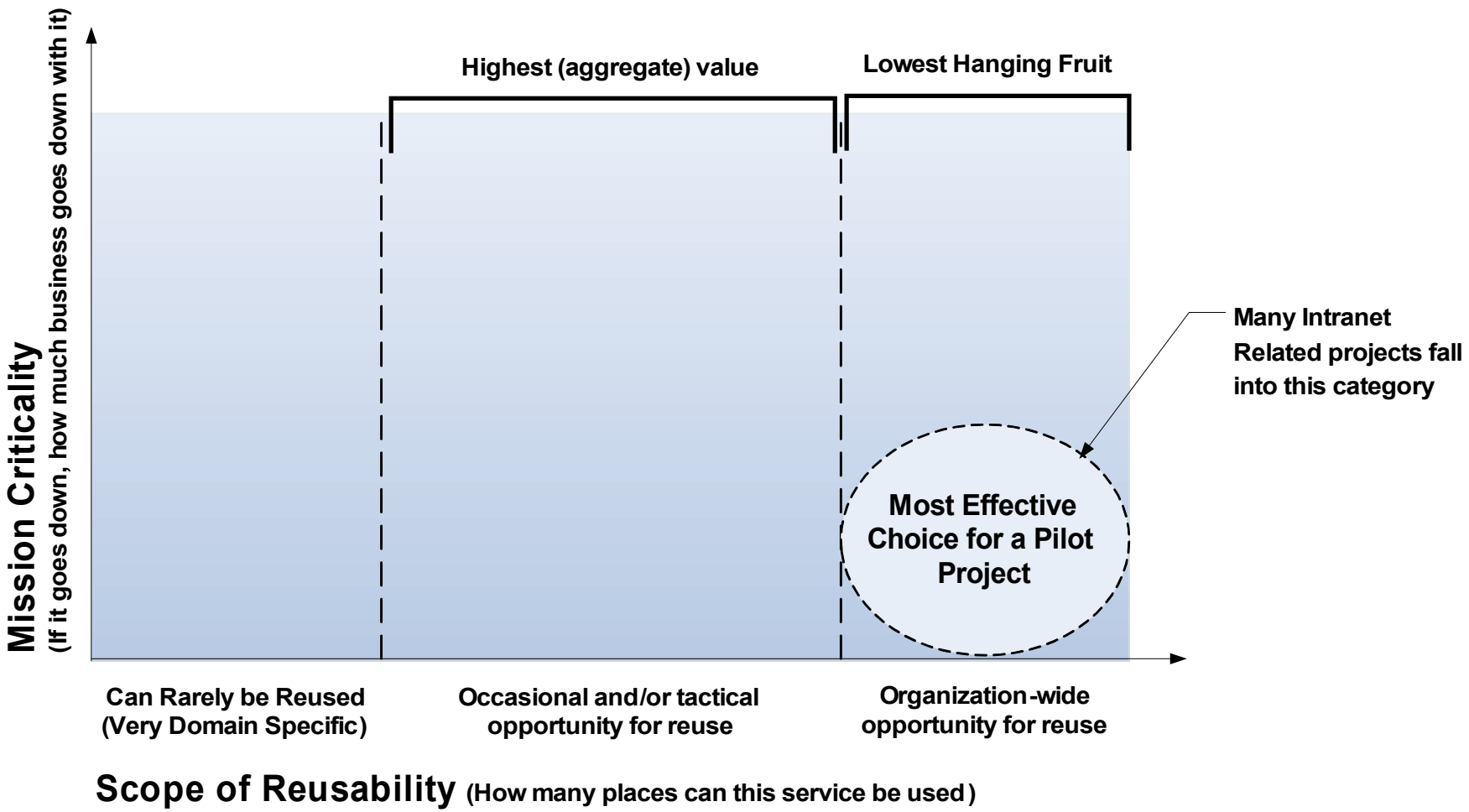
Lesson #5 : Web Services make implementing SOA easier, but they are not the same.



Lesson #6 : Service Oriented Integration is the most cost effective approach in the maintenance/evolution phase.



Lesson #7 : SOA pilot is an essential step in the early phase of the program.



Lesson #7 : SOA pilot – Critical elements & Success criterion

Critical elements of a SOA pilot

- **An architectural plan**
  - An SOA pilot, must take some subset of the SOA enterprise wide architecture plan & put it through it's paces.
- **Process Scope.**
  - It is best to start with a simple, yet important business process that can enable your SOA pilot to show a rapid, verifiable return.
- **Resource scope.**
  - Your approach to SOA should be both top-down (through process decomposition) as well as bottom-up (exposing existing functionality as fine-grained Services and composing them into coarse-grained business Services). For your pilot, it's important to identify those resources you wish to include in your wrapping and composing activities.
- **Iterative Methodology**
  - Start with a broad architectural plan, and then follow a methodology that enables you to dive into the process decomposition and Service definition activities, only to circle back, improve the plan, and revisit the processes and Services. Start small, but iterate quickly towards Services that show a real business return.

- **Clear acceptance criterion**
  - **Complete Architectural design**: Details about Service contracts, invocation styles, and other elements of a working SOA implementation.
  - **Reusable business services:**: Reusability for the Services you build, including the # of services & the number of consumers per service.
  - **Governance Criterion:** Pilot may lay out a framework for the governance infrastructure.

Lesson #8 : Finding the SOA champion for Business and IT alignment.

A champion is usually an individual, or sometimes a small team who understands the benefits of SOA, is able to communicate those benefits to both business and technical audiences, and who is committed to bringing SOA to their organization. Here are three viable choices:

- **Line of Business executive**
  - Few business executives have the technical depth and the architectural vision to champion SOA. The SOA message from these in-the-trenches executives carry a lot of weight with senior management. The true “meta-requirement” for SOA, after all, is to build an architecture implementation that addresses ever changing business requirements. Line of business must drive such a meta-requirement. Companies without such a champion, find themselves focusing on trees rather than the forest.

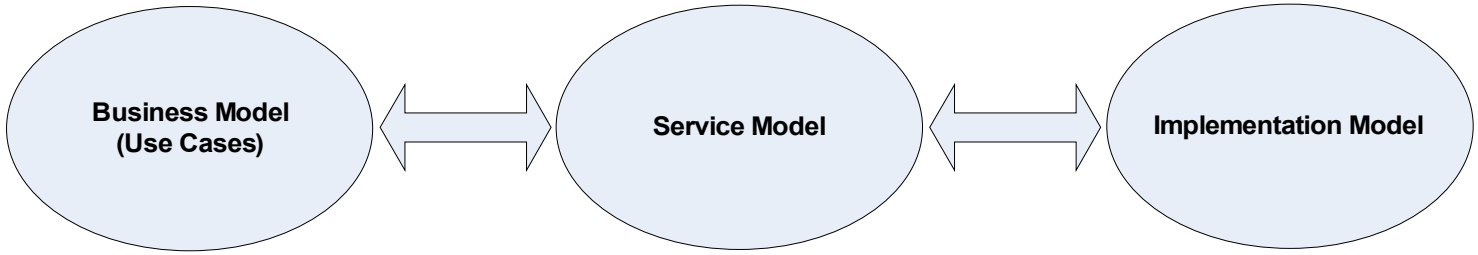
- **CIO**
  - CIO's have both business and technology responsibility, as well as the power and budget to make things happen. In practice, however, it is unusual for the CIO to be the champion due to lack of bandwidth.

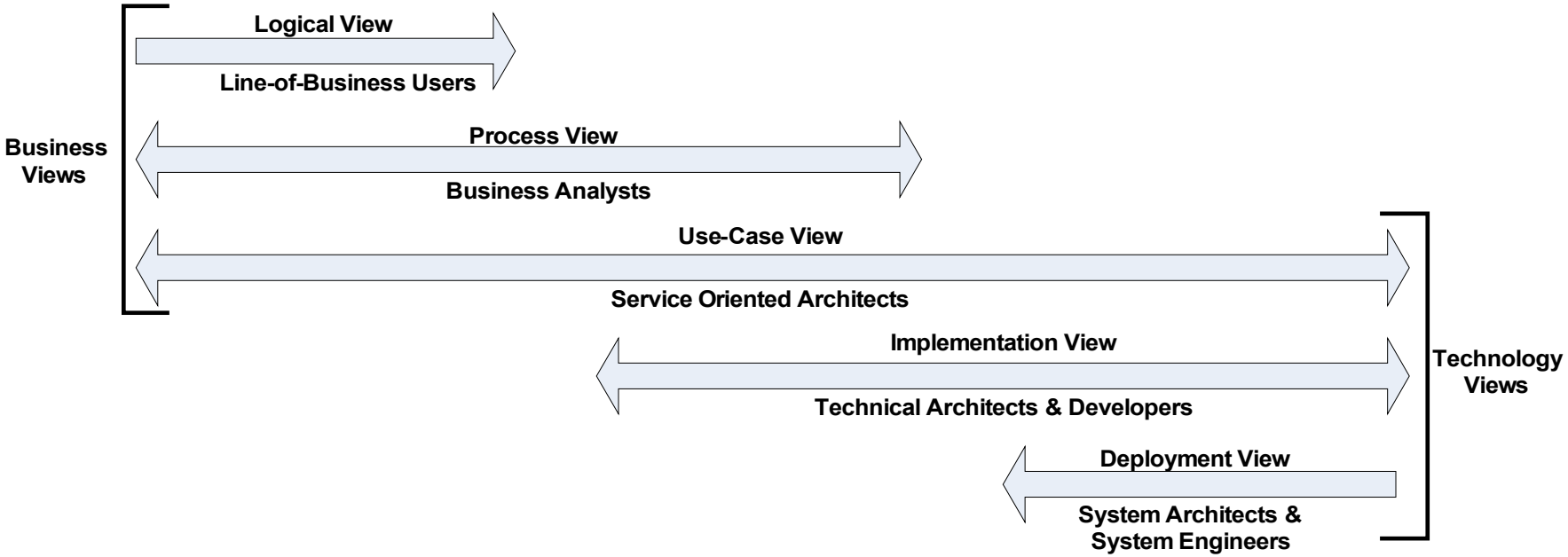
- **Chief Architect**
  - The chief architect typically is responsible for the overall corporate IT architecture strategy and how it meets the needs of the business. If this role has the budget and authority, then, is well positioned to be successful as a SOA champion.

Lesson #8 : Business and IT Alignment.

Maintain active and automatic feedback at all points of the Architecture

- Between business Models (Business Requirement and Process) and Service Model.
- Between Service Models and Implementation models.





Lesson #9 : You cannot buy SOA from a vendor.

Products that vendors have architected internally in a Service-oriented fashion may not help you in your Enterprise SOA initiative. Even so, there are many maturing products on the market that may truly contribute to the success of your SOA.

However, **you can buy the best SOA products on the market today, and you still won't have SOA.**

Buying the best tools won't make you a carpenter, after all.

Remember, **SOA consists of a set of best practices - - a discipline to follow, if you will.**

You should **partner with a SOA consulting firm**, but you cannot "buy SOA" from a vendor who doesn't offer SOA consulting as well.

Lesson #10 : SOA must be both (Top Down and Bottom Up)

Building SOA Top-Down (from scratch)	Building SOA Bottom-Up
❖ Create Architectural plans and detail design.	❖ Put service wrappers around existing applications.
➢ **Pros:** Agility, reuse and flexibility.	➢ **Pros:** Reduces cost of integration.
➢ **Cons:** May not be implement able, difficult to budget.	➢ **Cons:** May not be reusable, May be redundant, Management challenges.

SOA must be both (Top Down and Bottom Up)

- **Develop the vision (but not the details) ahead of time.**
- **Decompose some processes to identify target services.**
- **Build modest set of services.**
- **Compose applications to enable flexible processes.**
- **Refine architectural plans.**
- **R e p e a t – SOA SHOULD BE ITERATIVE**

❑ Enterprises who attempt to build all the pieces themselves without taking advantage of existing products on the market typically find themselves bogged down by the sheer scope of functionality necessary to get SOA right. Instead, companies should look for a combination of internal or external architectural expertise with an appropriate selection of software (and hardware) products that are well-suited to building SOA implementations.

Take a "leave and abstract" approach over "rip and replace".

Lesson #11 : Design Common Information model based on Industry standards – Adoption of Open Travel Alliance standards for Churchill.

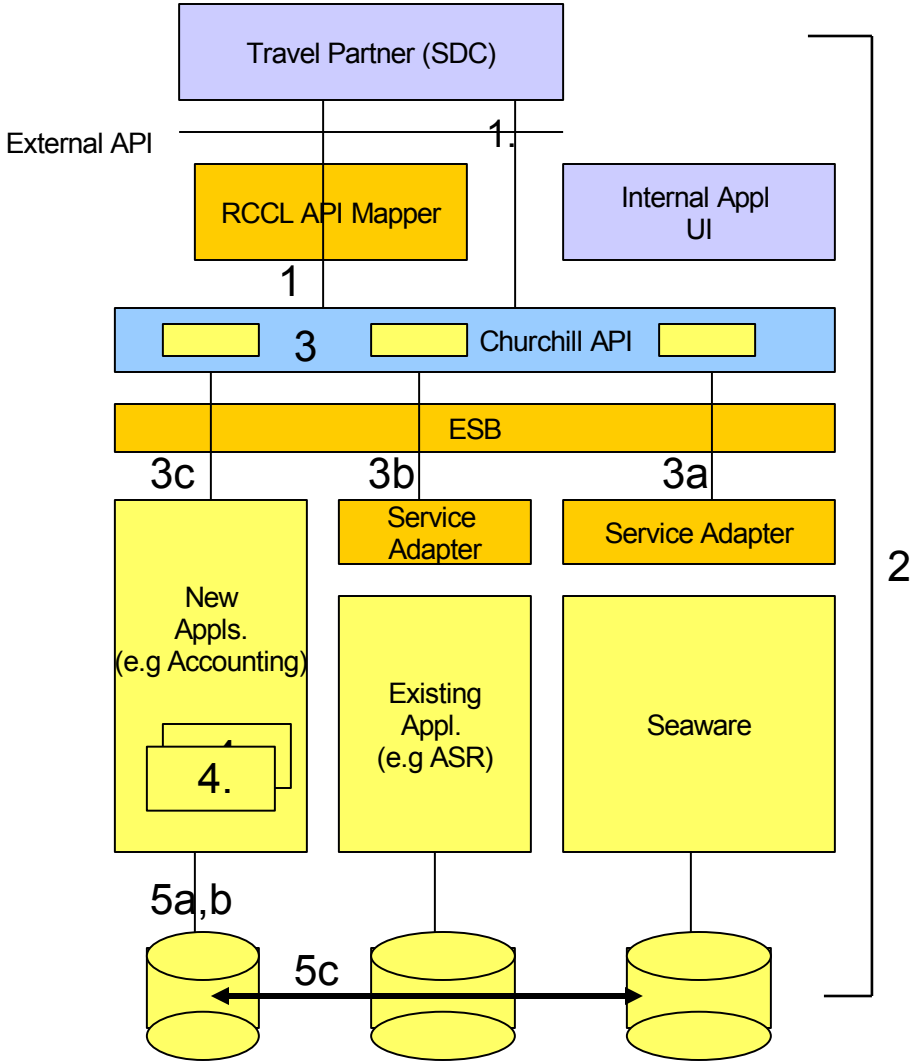
1. B-B Integration via OTA based API (external API) and migration from RCCL API

3. Standardize on RCCL common Information Model (Business Glossary/terminology)

5. Churchill Service Interfaces for **Interna**l Inter-application integration/communications
   - a. Seaware
   - b. "existing" applications
   - c. New applications

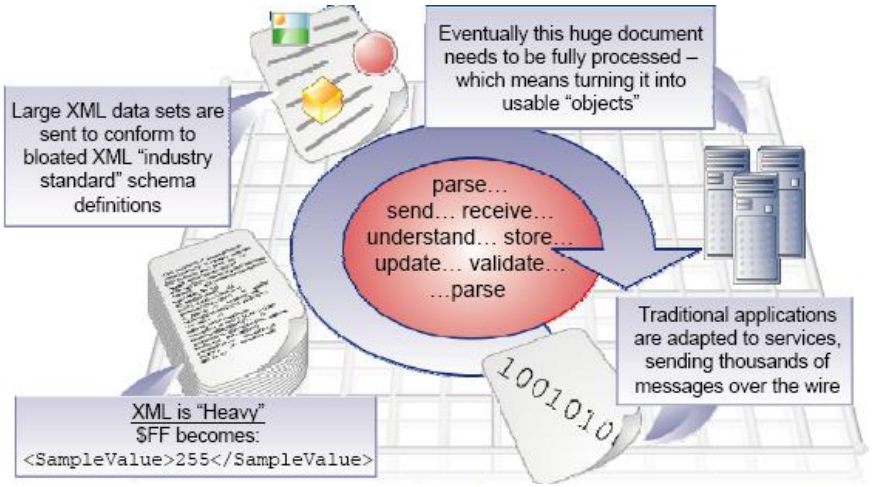
6. OTA XML datatypes map and extends to "New" Application Business Objects (BO) for specific programming languages e.g. Java

8. Integrate OTA Business Entities (XML schemas) with extensions as base for Logical Data Model (or Information model)
   - a. XML Schema Best Practices for naming conventions
   - b. XML & Relation DB mapping
   - c. Enterprise (Churchill level) data Integration



Lesson #12 :SOA Performance Issue: Use Hardware & Software Approaches as appropriate.

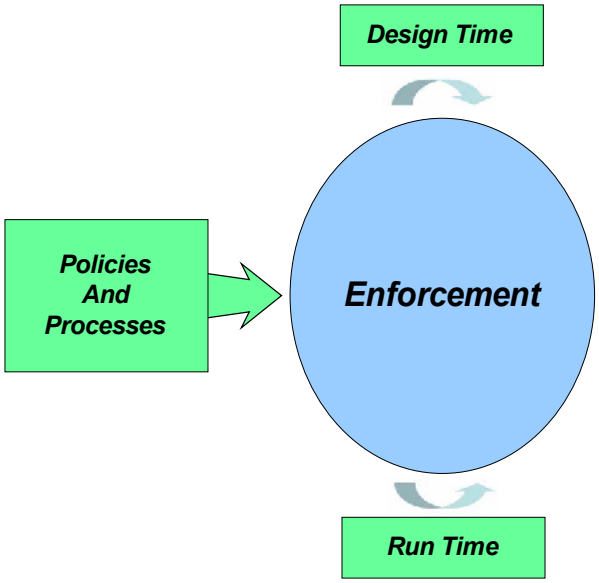
- **SOA requires more metatdata than ever …..**
  - Web Services-based SOA is mostly XML-based.
  - XML is text heavy and verbose.
- **The XML Processing challenge.**
  - Per-message receive, authenticate, decrypt, validate, canonicalize, map to memory models, transform, parse, aggregate, sign, encrypt, transmit.
- **High volumes of messages and very large messages two separate problems.**

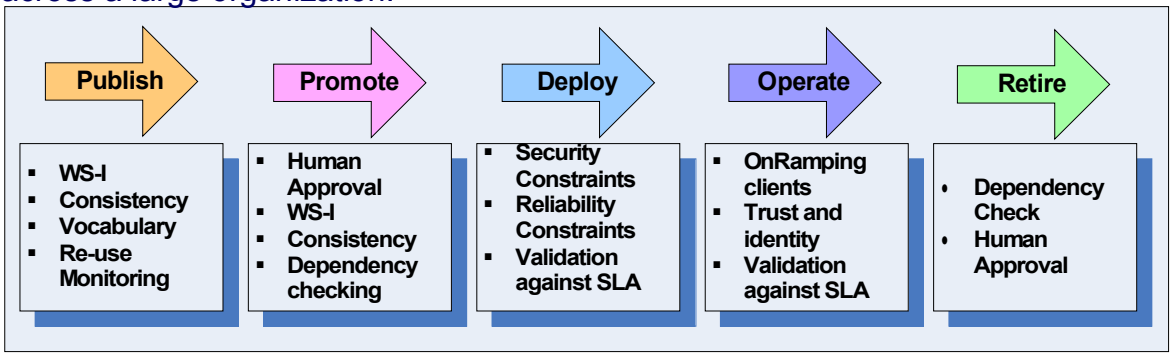


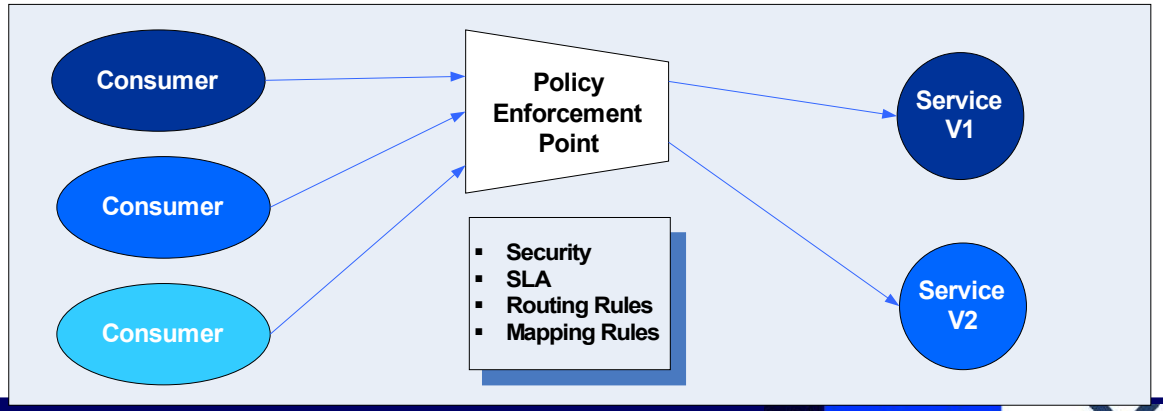
Hardware	Software
❖ **PROS:** ➢ Dedicated Processing Power ➢ Control of variability. ➢ Hardened OS ➢ Simple to install and maintain.	❖ **PROS:** ➢ Developers in control. ➢ Easy to customize and upgrade.
❖ **CONS**: ➢ Must Go into the data center. ➢ More difficult to upgrade. ➢ Limited customizability.	❖ **CONS:** ➢ Harder to install and maintain. ➢ Relies upon hardware for performance. ➢ Too much variability in configuration.

Lesson #13 : SOA Governance is required in all phases of SOA lifecycle.

- SOA Governance can enable organizations to maximize the business benefits of SOA, which can include **increased process flexibility**, **improved responsiveness** and **reduced IT maintenance costs**.
- It also mitigates many of the business risks inherent in SOA adoption by **establishing decision rights**, **guiding the definition of appropriate services, managing assets for reuse and measuring effectiveness**.
- Setup a **Center of Excellence (COE)** to initiate SOA Governance
  - SOA experts who maintain knowledgebase of best practices.
  - Drive SOA policies and processes.
  - Can unify approaches across a large organization.







Summary – 13 Lessons

1. SOA adoption is a gradual process.
3. Focus on how to do SOA right instead of getting lost in the Alphabet soup.
5. Do Not approach SOA like Traditional distributed computing.
7. SOA is not the only choice for every project.
9. Web Service is not the same as SOA.
11. SOI is the most cost effective integration approach in the long run.
13. SOA Pilot is an essential step in the early (technology adoption) phase of the program.
15. Business to IT alignment – SOA Champion, Governance of Service Model.
17. You cannot buy SOA from a vendor.
19. Approach SOA Top down and Bottom up.
21. Design Common Information Model based on industry standards.
23. SOA Performance issue – use hardware and software options as appropriate.
25. SOA Governance is required in all phases of the SOA lifecycle.

Thank You

Contact Information:

Prashant Naik,
IT – Chief Engineer,
Royal Caribbean Cruise Lines.

Email: pnaik@rccl.com

References: ZapThink,
IBM Developerworks.