Case Study: Making the Business Case for a Transition to SOA

Max Dolgicer
Managing Director
mdolgicer@isg-inc.com

Gerhard Bayer
Senior Consultant
gbayer@isg-inc.com

International Systems Group (ISG), Inc.
250 West 57 Street, Suite 2532
New York, NY, 10107
Tel: 212 489 0400
Fax: 212 489 1125
isg@isg-inc.com
http://www.isg-inc.com

International Systems Group (ISG), Inc.

- Founded on the premises of Middleware technologies 16 years ago in New York City
- Three practice areas
  - IT Strategy and SOA Architecture practice
    - Definition of SOA
    - SOA Assessment and SOA Compliance Verification
    - CIO Summits / IT and Business Objectives Alignment
  - SOA Modeling, Design and Implementation practice
    - Development of applications and infrastructure (i.e. foundation) services
    - EAI solutions for A2A and B2B integration
  - Education and Training practice
    - Public and on-site training
    - Out-of-the-box and customized classes
    - Defining and Implementing Service Oriented Architectures (SOA) – From Theory to Practice
    - Implementing Service Oriented Applications – An Introduction to Service Oriented Analysis, Modeling and Design

International Systems Group (ISG), Inc.

- ISG professionals average 15+ years of experience in development and integration of distributed systems
  - Real hands-on experience spans end-user corporations and software vendors

Agenda

- Business and technical objectives of the transition to SOA
  - How business objectives are addressed by IT
- How SOA is being implemented at Carey
  - Service layers
  - Architecture overview
  - Service reuse for B2B
- Facilitating alignment of IT with new business strategies
  - Service reusability enables business agility
- Return on investment (ROI) calculation

Case Study: Chauffeured Services Company

- Carey provides premium limousine services
- The clientele of Carey International comprises
  - Individual customers including many celebrities
  - Corporate clients
  - Travel agencies
- Carey provides services to its customers via
  - Carey owned subsidiaries
  - Affiliates
  - Licensees and farm outs
- In addition, there are external sources of business information
  - E.g. Global Distribution Systems (GDSs)

Evolution Of Carey IT

- IT at Carey subsidiaries consists of locally installed and managed applications
  - Legacy software (CSW)
    - Manages reservations, dispatches and end-of-job functions at local subsidiaries and franchises
  - Hudson software - third party software, same as CSW but includes also billing functions
  - Odyssey - third party software
- IT at Carey Corporate Headquarters consist of
  - CESRes – home-grown, distributed applications
    - Provides reservation, profiles and customer service functions
    - **Centrally deployed in the data center**
    - Now WebSphere J2EE based
      - Originally developed as Java/Corba and ported to WebSphere
    - Supports multiple inbound channels
      - www.ecarey.com
      - Call center where rich client (i.e. Swing) is used

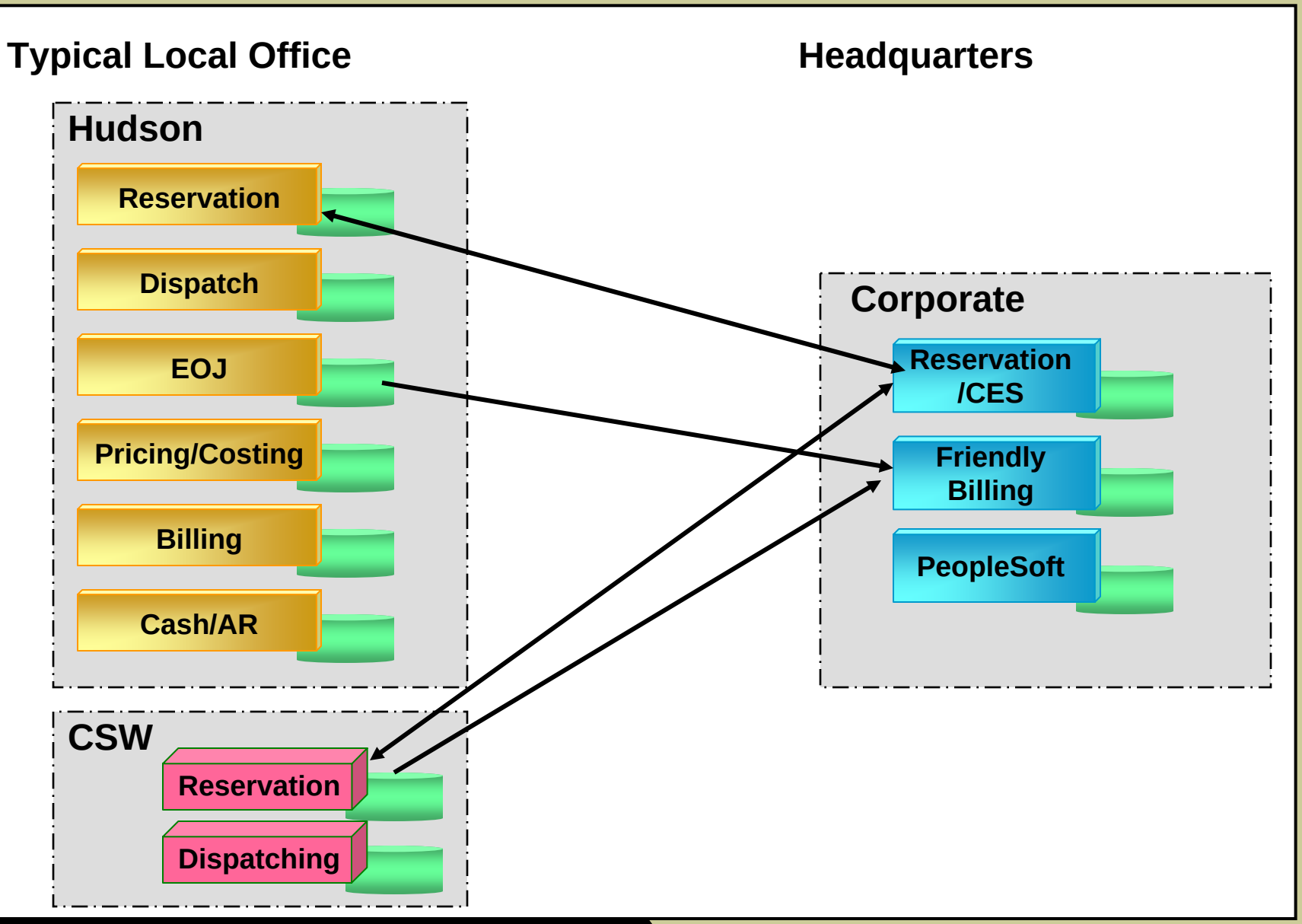
Evolution Of Carey IT

- "Friendly Billing"
  - Provides generation of invoices
  - **Centrally deployed**
  - Legacy, 4GL (i.e. Magic based) system
- PeopleSoft to perform all the AR/AP functions
  - **Centrally deployed**

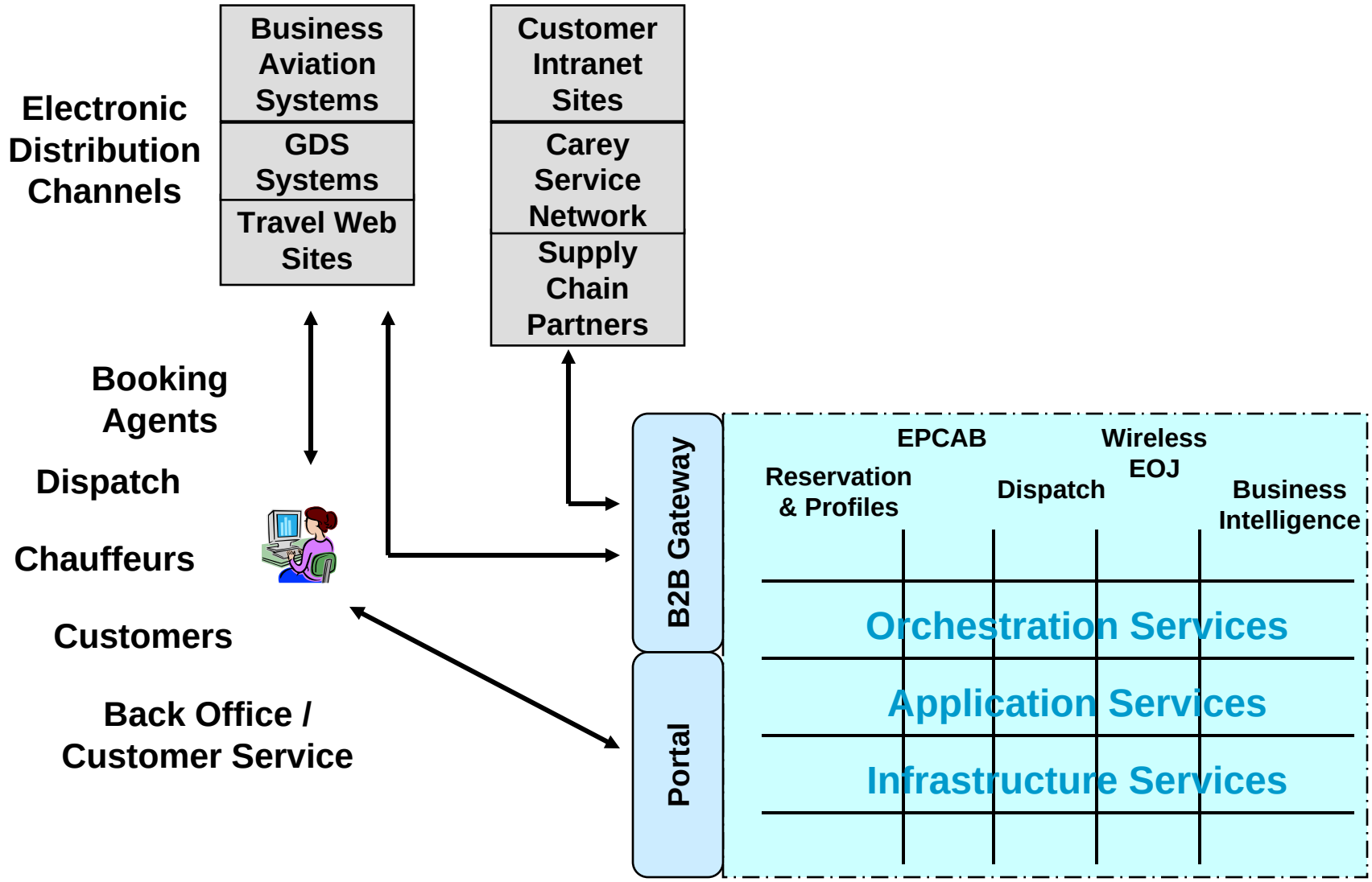
Carey Business Requirements For IT

- This “accidental architecture” has hampered the business
  - About 80% of reservations are made at one of the local offices
    - Creates a huge integration problem and a lot of manual labor especially when it comes to handling exceptions
  - Only about 20% reservations are made at the call center or via www.ecarey.com
- The goals:
  - Support all of the core business functionality from one (logically) central location
  - Support the expansion of service offerings
  - Support the acquisition of other companies

Old "Accidental" Architecture



Target New Architecture



Important Book

- Service Oriented Architecture Concepts, Technology, and Design by Thomas ERL ISBN 0-13-185858-0

Business Objectives & IT Approach

Carey Business Objectives	Addressed by IT
Revenue Enhancement *New service offerings* *Cross-selling* *Acquisitions*	• Better leverage of the IT infrastructure and processes • Increased extensibility and adaptability of the core applications • Avoid one-off integration solutions
Customer Service Enrichment *Single view if the customer* *Better customer self-service* *Improved business process consistency*	• Creation of a centralized reservation system with customer profiles • SOA enables efficient integration of a portal with back office systems • Centralized SOA leads to improved business process consistency
Allow travel agents to book with Carey using major GDS systems.	• Automated B2B integration with major GDSs and other types of business partners.
Total cost of ownership (TOC)	• Service reuse, decreased complexity
Risk management	• Core services (foundation and some application) are implemented by highly skilled designers/developers

General Objectives Of The Carey SOA

- The main objectives of the Service Oriented Architecture at Carey include:
  - Contain cost
    - Maximize the efficiency of services by eliminating redundancies and increasing reusability
    - Reduce the number of applications from 35 to 15
    - Decrease in software license cost & maintenance complexity
  - Create a comprehensive “big picture” view of the enterprise systems
    - Old services can be migrated
    - New services can be created without significant risk or investment of additional resources
  - Introduces a high level of accountability, trace-ability, and accuracy to the application development process

Architecture Scope

- The overall goal is to define a central system that delivers full business functionality to all Carey sites, but that is maintained in a single location.
- The scope of the architecture includes
  - Consolidation of the reservation system
    - ***Migrate*** to the new J2EE environment
    - ***Encapsulate*** existing services
    - ***Surround*** by new services
    - ***Replace*** legacy implementation
    - ***Expand*** the reach: more and more reservations will be handled by the central reservation system
  - Really “surround and retire” as oppose to “rip and replace” or “leave and layer” strategy

Architecture Scope

- The scope of the architecture includes
  - Replacement of the billing system (i.e. EPCAB)
    - Current: central legacy 4GL system (i.e. Friendly Billing)
    - Some subsidiaries manage billing locally, others use a manual process to transfer billing to the central system
    - The new system will handle pricing, costing, and adjustments centrally
      - End-Of-Job information will be sent from subsidiaries to the central system
      - Drivers can send adjustments including expenses directly from their J2ME enabled handheld devices
  - Replacement of the local dispatching applications by a new central dispatching system
    - SLA for this application was a big challenge
  - Common infrastructure (i.e. foundation) and common application services

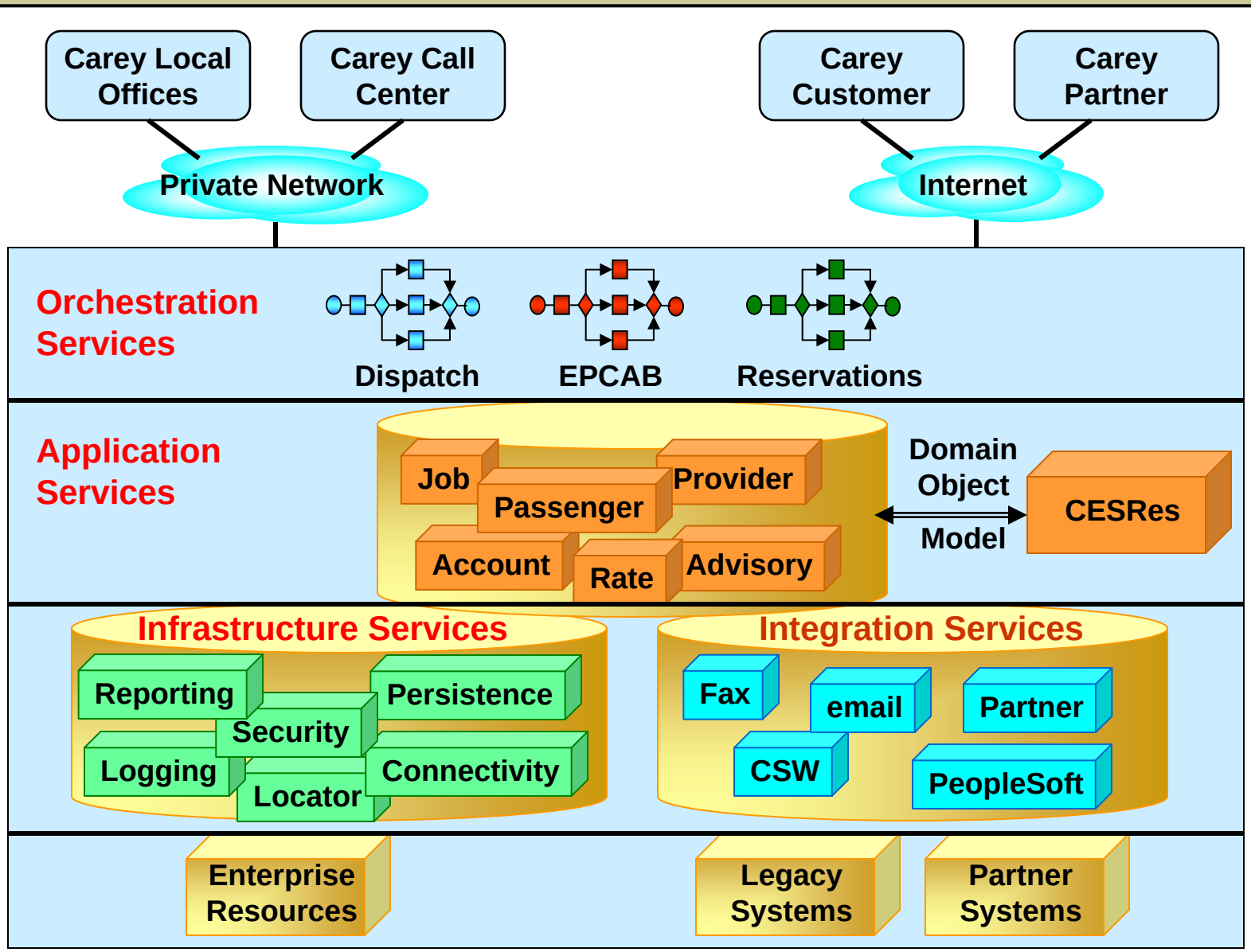
The Layered Services Architecture

- The SOA has been developed following best practice guidelines for service analysis, modeling, and design
- Key architectural principles that were applied are the separation of concerns and design for reusability
  - Separation of concerns leads to a separation of service functionality into layers
  - Specifically, the SOA has been delineated into the following service layers:
    - Orchestration Services layer
    - Application Services layer
    - Infrastructure (i.e. Foundation) Services layer

The Layered Services Architecture

- The breakdown into distinct layers facilitates decoupling of the services
  - One of the most important characteristics of a well-defined SOA
- Reusability is achieved by designing key application services and infrastructure services that are autonomous and are agnostic of the business process context within which they are executed
  - For example, the services are being utilized by different presentation layers
    - Web interface (customer portal)
    - Thick client (Swing interface for internal users)
    - B2B channels (business partners)

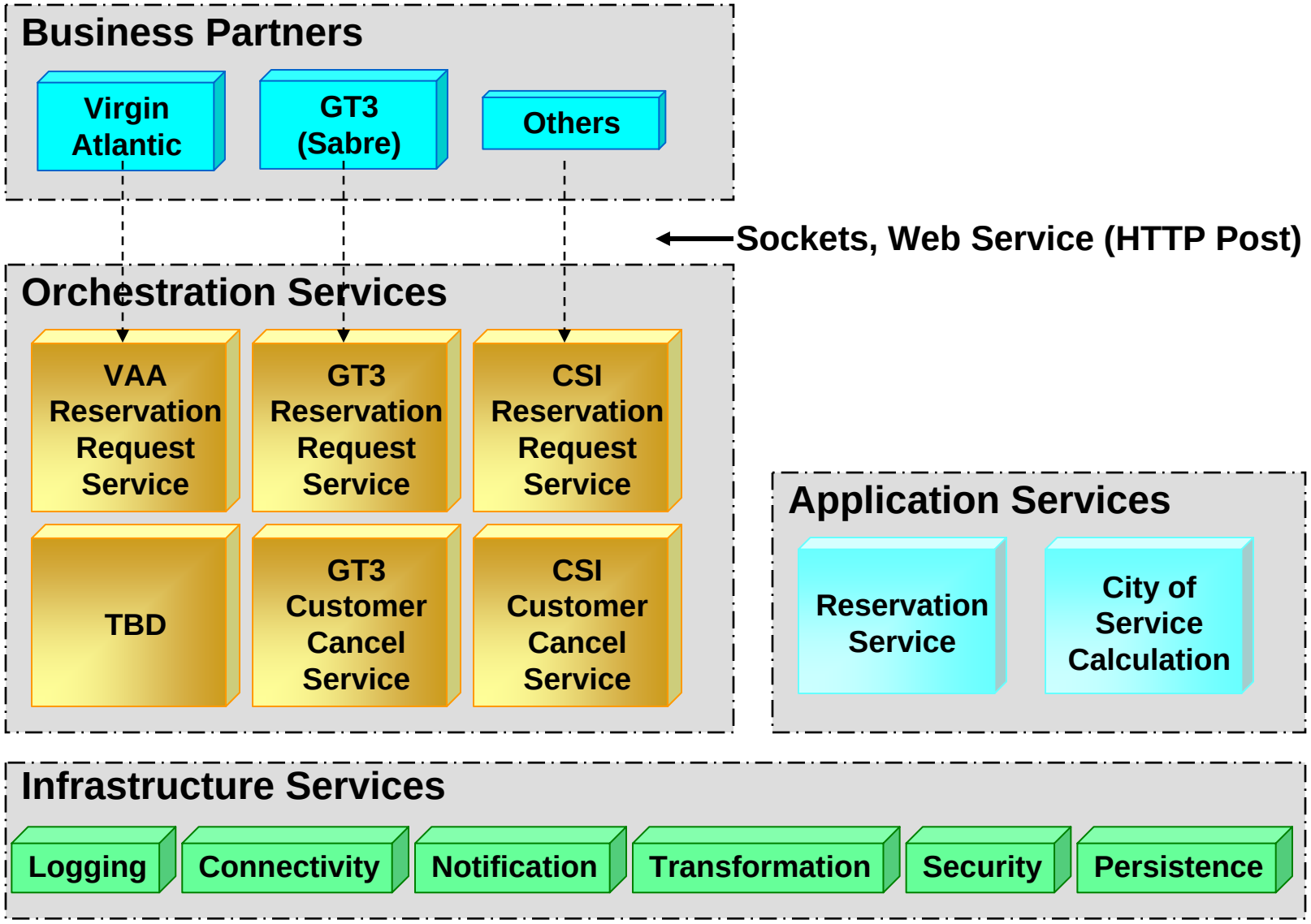
Architecture Overview



Carey's SOA Foundation Core

- The SOA prescribes a multi-tier architecture foundation
  - Presentation tier
    - Subsidiaries, call center, customers, business partners
  - Business process tier
    - High-level, adaptable business logic
  - Core business logic tier (i.e. Application Services)
    - Reusable domain logic, independent of business processes
  - Infrastructure tier
    - Insulates applications from low-level system services
  - Back-office tier
    - ERP, legacy systems
- This foundation supports all business applications and a B2B Gateway

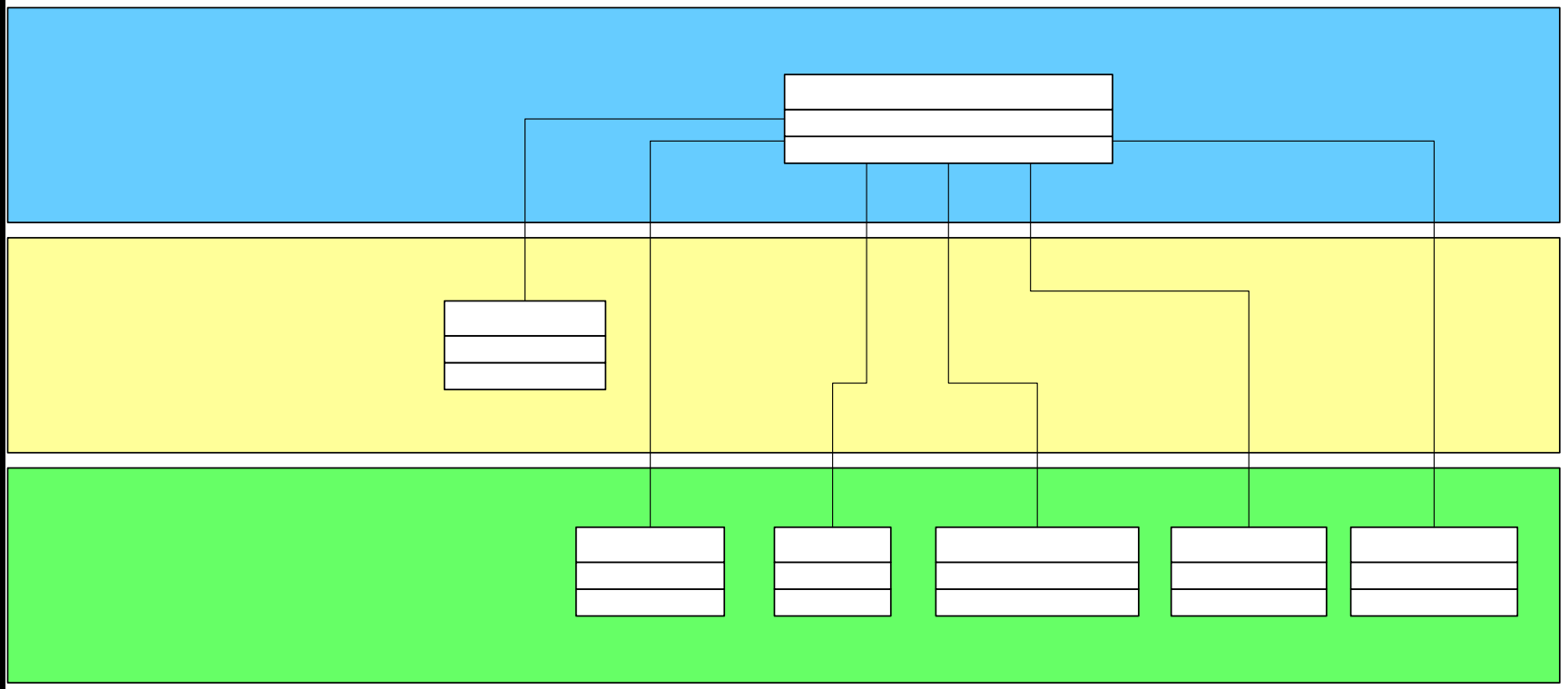
B2B Gateway Service Layers



The Layered Services Architecture - Example

- A multitude of business partners interact with Carey through different B2B channels
- One partner is the Ground Travel Technology Team (GT3)
  - It provides an interface to the major computerized reservation systems
    - E.g. Sabre, Apollo and Worldspan
  - Reservation requests are being sent from GT3 to Carey

The Layered Services Architecture - Example



Orchestration Services

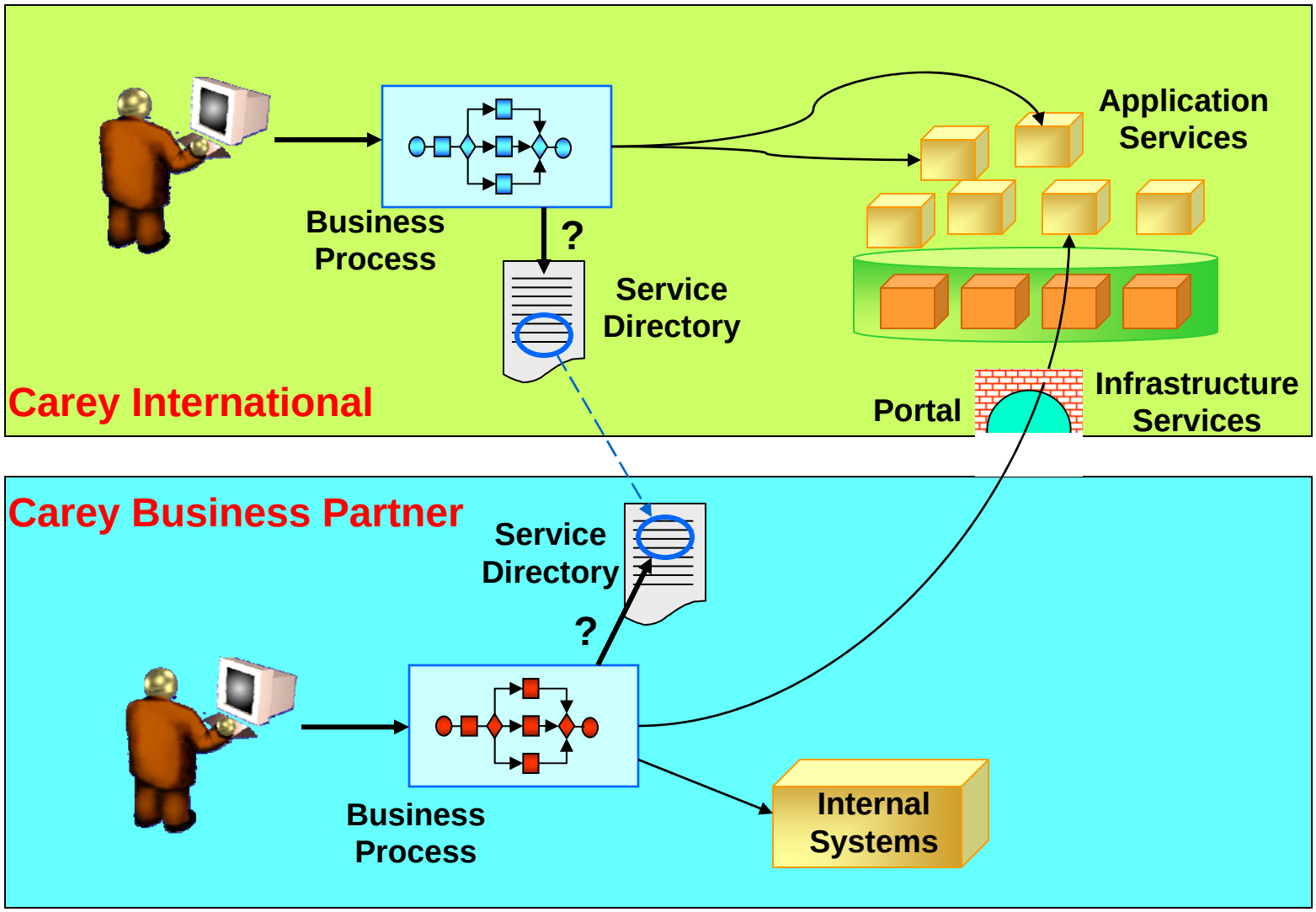
The Layered Services Architecture - Example

- The GT3 Reservation Request service model implements a business process where the GT3 system sends a request to Carey to create a new reservation, change or delete an existing reservation adhering to some well defined business rules
- An Orchestration Service manages the flow of business transactions between Carey and GT3
  - It also maps the reservation requests to activities that are internal to Carey
  - It uses common infrastructure and application services
    - Reservation, Logging, Security, Transformation, Notification, and Connectivity

How To Align Business And IT Through SOA

- Most companies face an increasing complexity of their enterprise application portfolios
  - Fulfilling business demands by adding new applications and packages
  - Building more connections between systems in order to achieve integration
  - Portfolio complexity slows down responsiveness of IT to business requests and in turn negatively affects business agility
- SOA allows to streamline application portfolios
  - Reduce redundancies
  - Simplify connectivity across internal and external enterprise boundaries through standardization

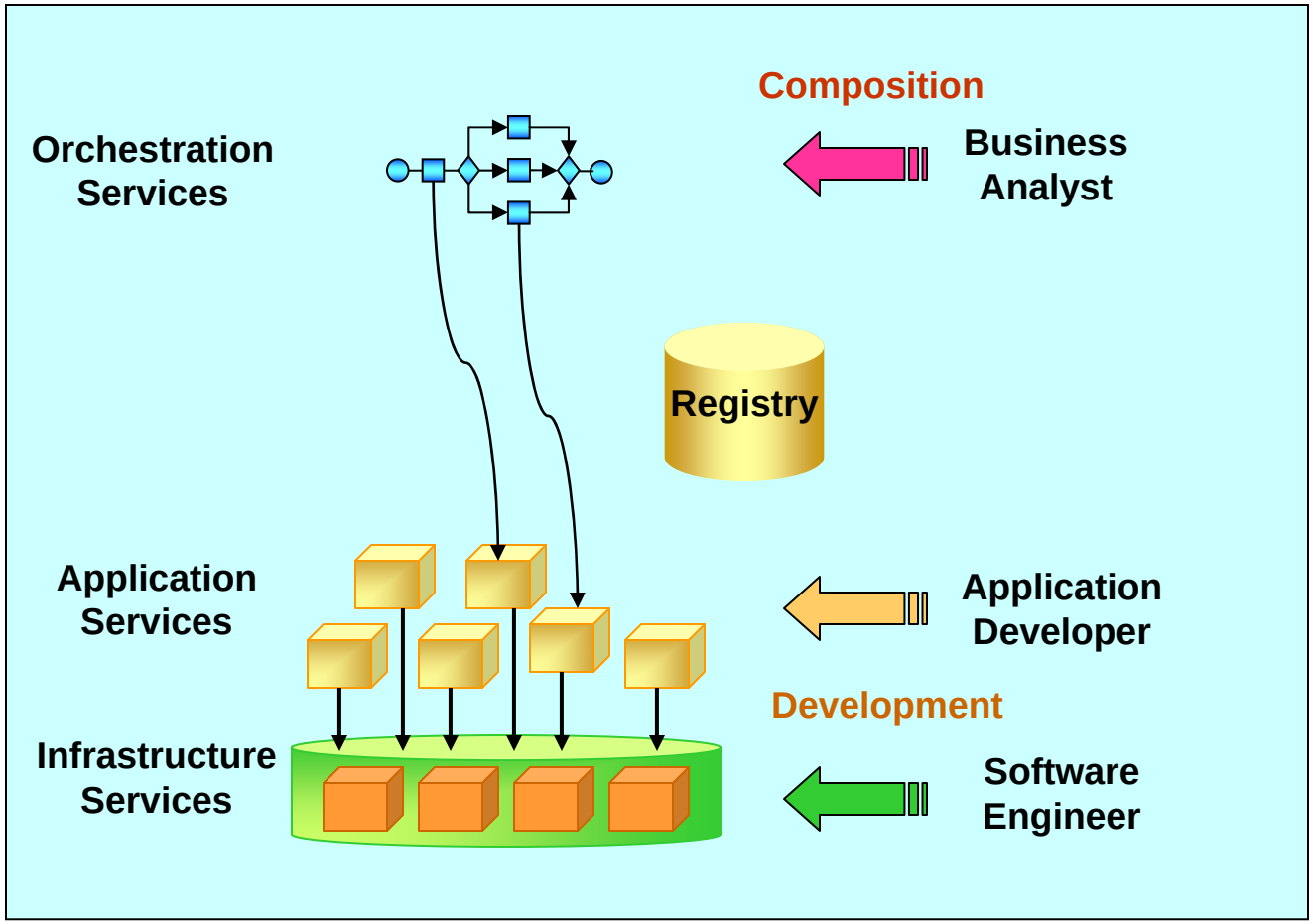
Service Reuse = Business Agility



Service Reuse = Business Agility

- Across a number of projects, more and more application services and infrastructure services are implemented
  - Build-out of service repository
- New business processes are implemented as new orchestration services
  - Potentially, additional application and infrastructure services need to be developed
    - **Develop as you go approach**
- Services implemented for use by internal processes can be reused for B2B
  - External partners do not utilize specific services, but typically connect to exposed orchestration services

Process Composition = Business Agility



Process Composition = Business Agility

- The SOA facilitates a separation of concerns
- Therefore the skill sets of the developers can be separated
  - Software Engineer
    - Implement infrastructure services that require experience with system software, e.g. persistence service, security service, scheduling service, reporting service
    - Develop design guidelines for application services that ensure non-functional requirements are met (e.g. clustering, fail-over, SLA)
  - Application Developer
    - Possess good business domain knowledge and develop application services that fulfill functional requirements
  - Business Analyst
    - Assemble application services into new business processes through the definition of orchestration services

ROI Model For Software Development

- The number of development hours saved by reusing a software component is the basis for calculating ROI
- Assumptions:

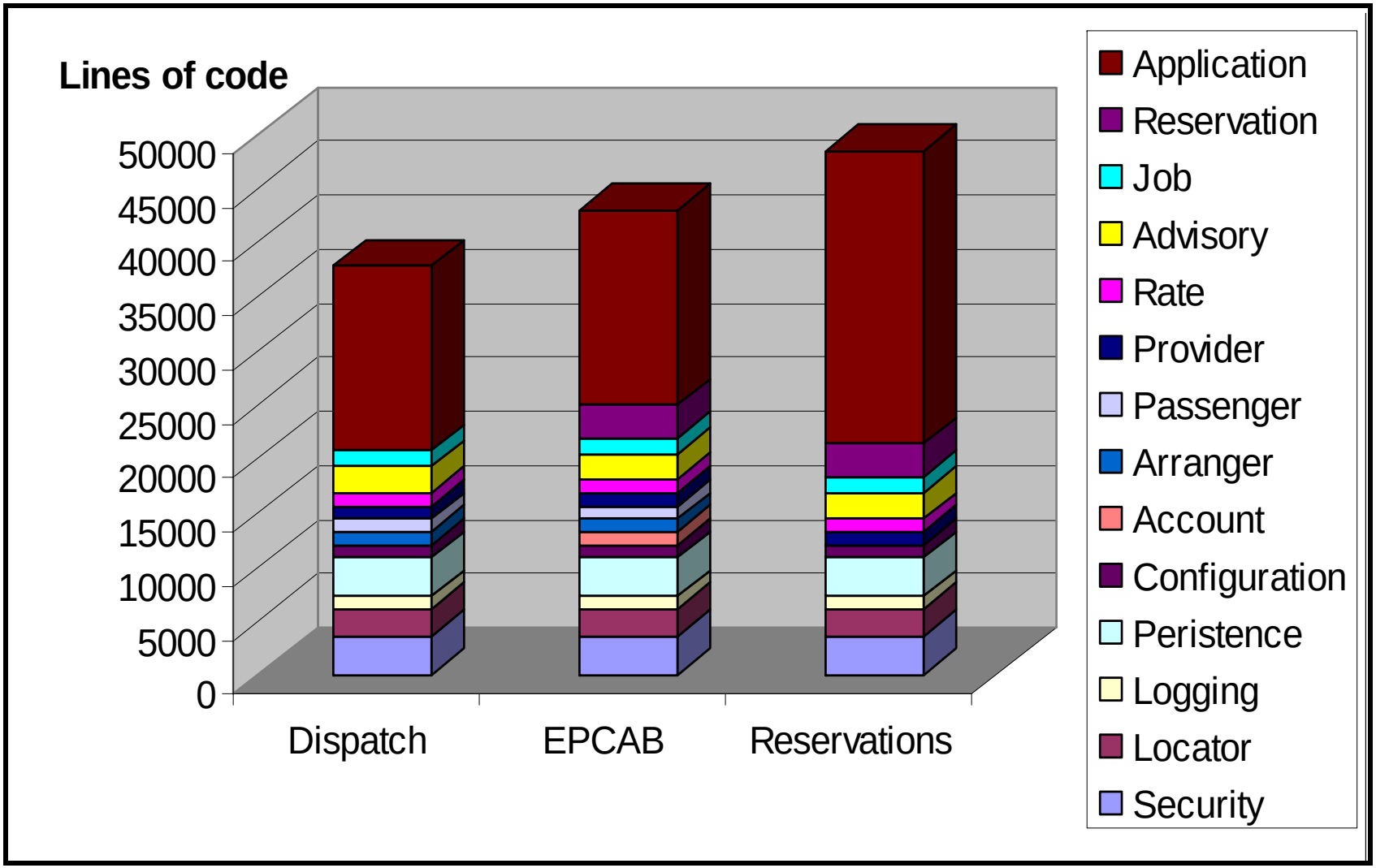
Development cost per line of code	$30
Development cost per line of ***reusable*** code	$40
Cost of reusing code per line	$5
Number of defects per 1,000 lines of code	1
Cost of defect repair (post production)	$5,000

Based on Jeffrey Poulin, "Measuring Software Reuse: Principles, Practices, and Economic Models."

Example of ROI Calculations

Initial Investment	Infrastructure Services	Application Services		
Lines of code built for reuse	12,000	15,000		
Cost	**$480,000**	**$600,000**		
Total initial cost		$1,080,000		
Application projects	**Dispatch**	**EPCAB**	**Accumulated**	**Reservations**
Total lines of code in project	40,000	45,000	85,000	50,000
Reuse factor	0.9	1.0		0.9
Lines of code reused	22,950	27,000	49,950	22,950
Lines of code built from scratch	17,050	18,000	35,050	27,050
Cost without reuse	$1,200,000	$1,350,000	$2,550,000	$1,500,000
Cost with reuse	$626,250	$675,000	$2,381,250	$926,250
Savings	**$573,750**	**$675,000**	**$168,750**	**$573,750**
Defect repair cost without reuse	$200,000	$225,000	$425,000	$250,000
Defect repair cost with reuse	$85,250	$90,000	$175,250	$135,250
Savings	**$114,750**	**$135,000**	**$249,750**	**$114,750**
Total savings	$688,500	$810,000	$418,500	$688,500
Savings percentage			16%	46%
ROI			39%	64%

Code Reuse



INTERNATIONAL Systems Group, Inc.

From Chaos To Order - Delivering e-Business Integration Solutions

INTERNATIONAL
Systems Group, Inc.

250 West 57 Street,
Suite 2532
New York, NY 10107

Tel: (212) 489-0400
Fax: (212) 489-1125

email: isg@isg-inc.com
http://www.isg-inc.com