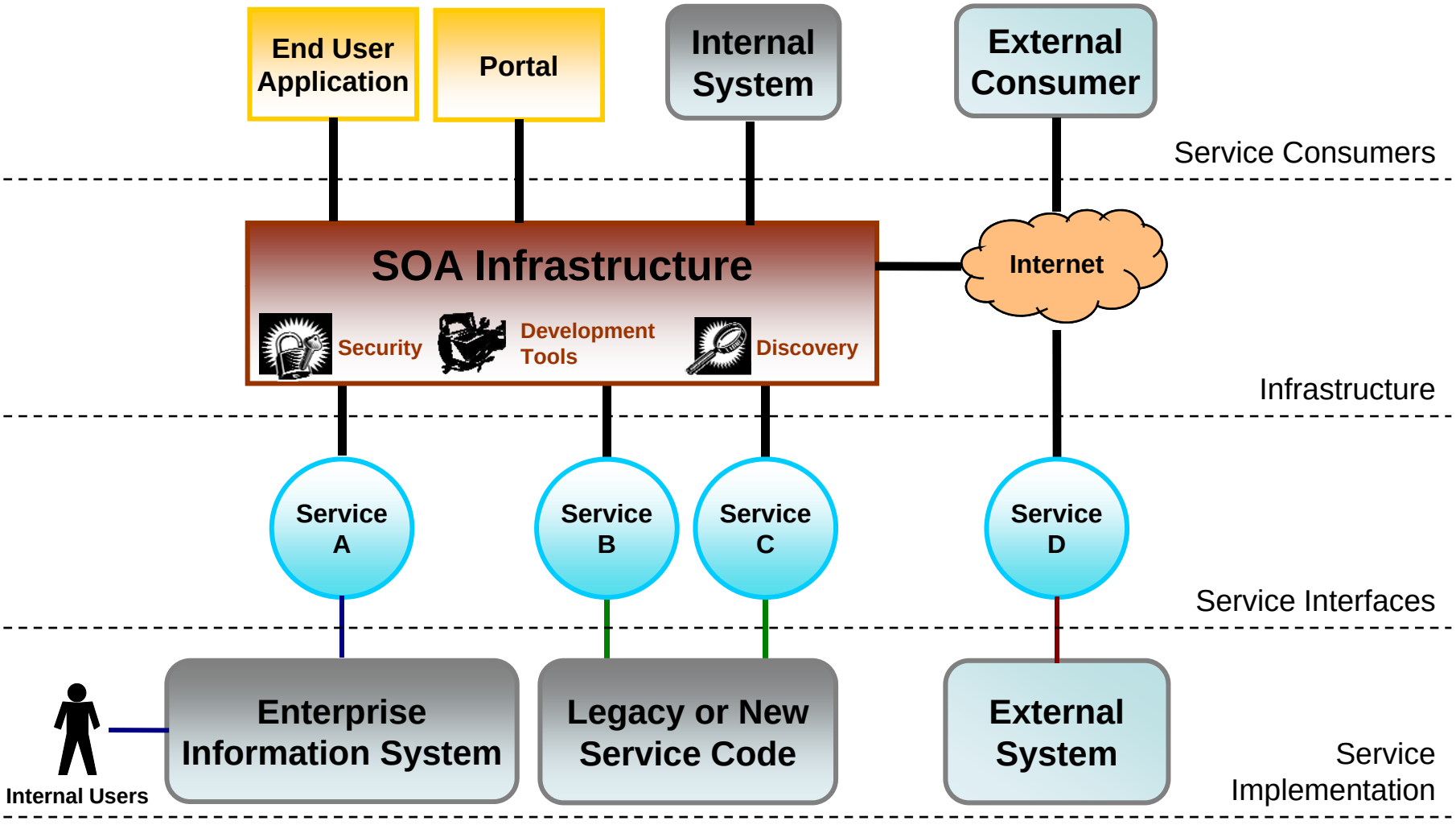
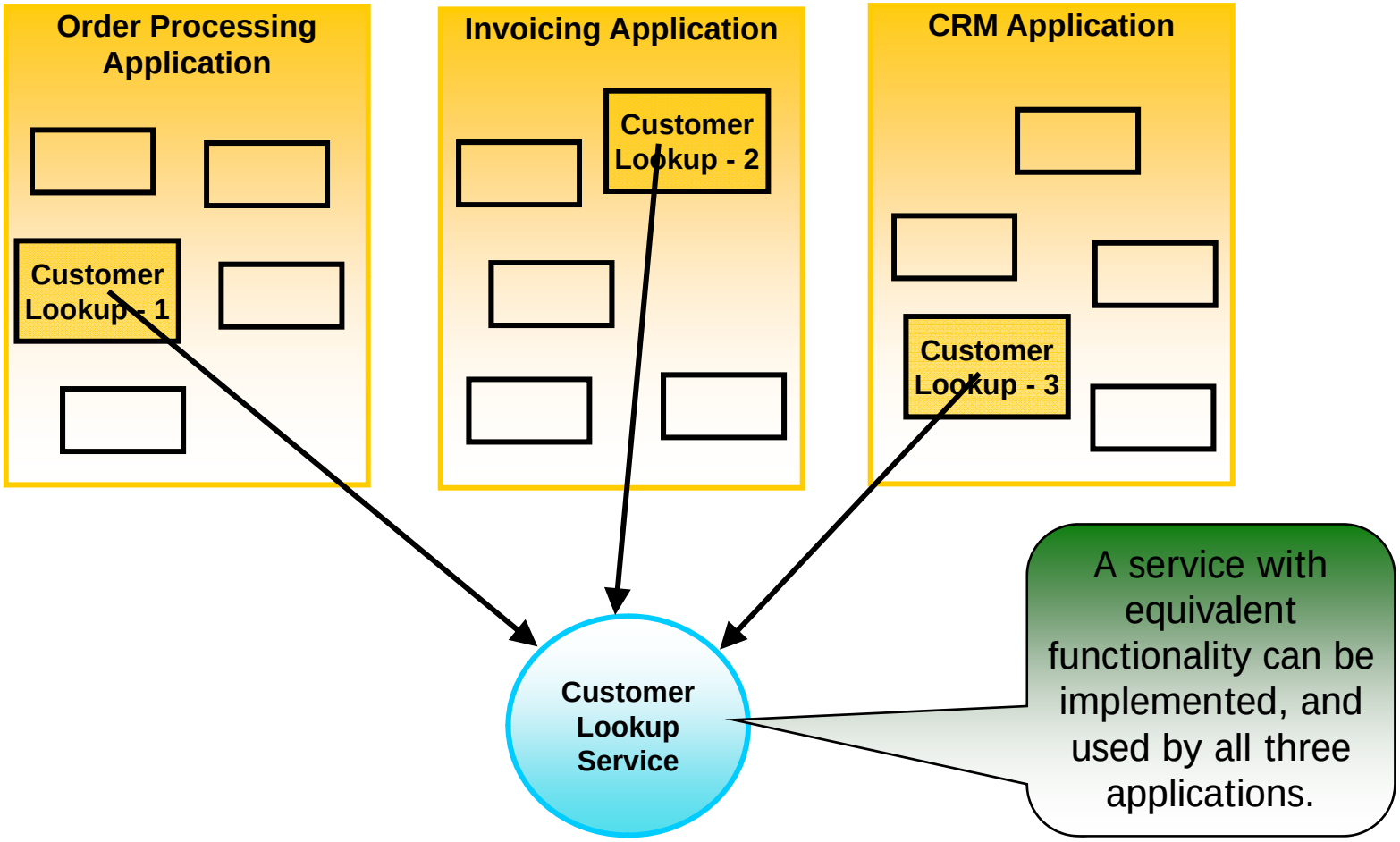
Tradeoffs in Using SOA with Legacy Components

Scott Tilley
stilley@sei.cmu.edu

Components of an SOA-Based System



Realizing Cost-Efficiency through Services



All Legacy Systems Can Be Easily Integrated into an SOA Environment … NOT!

Upfront hands-on analysis on the technical feasibility and return-on-investment must be performed to avoid last-minute surprises.

- Is it technically feasible to create a service from the legacy system or part of the system?
- How much would it cost to expose the legacy system as services?
- Is this cost plus the cost of maintaining the legacy system more than the cost of replacing it with a new one?
- What changes will have to be made to the legacy system?
- How much will these changes affect current users and other production systems?

It might just not make sense to migrate the legacy system to an SOA environment.

Legacy System Challenges

Poor separation of concerns

- User interface code tightly coupled with business function code

Tool availability

- Target is Web Services; XML / SOAP libraries not available for all legacy platforms.

Architectural mismatch

- Asynchronous call to the service conflict with legacy system synchronous behavior.

Operational mismatch

- Legacy system is batch-oriented, the service user expects an immediate response.

Process mismatch

- Legacy system maintenance done differently than SOA system development.

Dependencies on commercial products

- Licensing issues?

Addressing Legacy System Challenges

Identify relevant and non-relevant legacy components.

- Not all legacy components can be meaningfully reused as services—from a strategic and a technical perspective.

Make decisions based on “hands-on,” contextual analysis.

- System-specific analysis is important because every system is unique.
- Previous analysis and results can be used a guidelines.
- Use T-Checks for small, representative, case studies

Estimate cost, risk, and confidence of estimates of changes required to each legacy component.

Service Migration and Reuse Technique

SMART analyzes the viability of reusing legacy components as the basis for services by answering these questions:

- Does it make sense to migrate the legacy system to services?
- What services make sense to develop?
- What components can be mined to derive these services?
- What changes are needed to accomplish the migration?
- What migration strategies are most appropriate?
- What are the preliminary estimates of cost and risk?

Three Elements of SMART

Process 	Service Migration Interview Guide (SMIG) 	Artifacts 
Gathers information about • Goals and expectations of migration effort • Candidate services • Legacy components • Target SOA environment Analyzes gap between legacy and target state	Guides discussions in initial SMART activities	• Stakeholder List • Characteristics List • Migration Issues List • Business Process-Service Mapping • Service Table • Component Table • Notional SOA-Based System Architecture • Service-Component Alternatives • Migration Strategy

SMART Process Activities