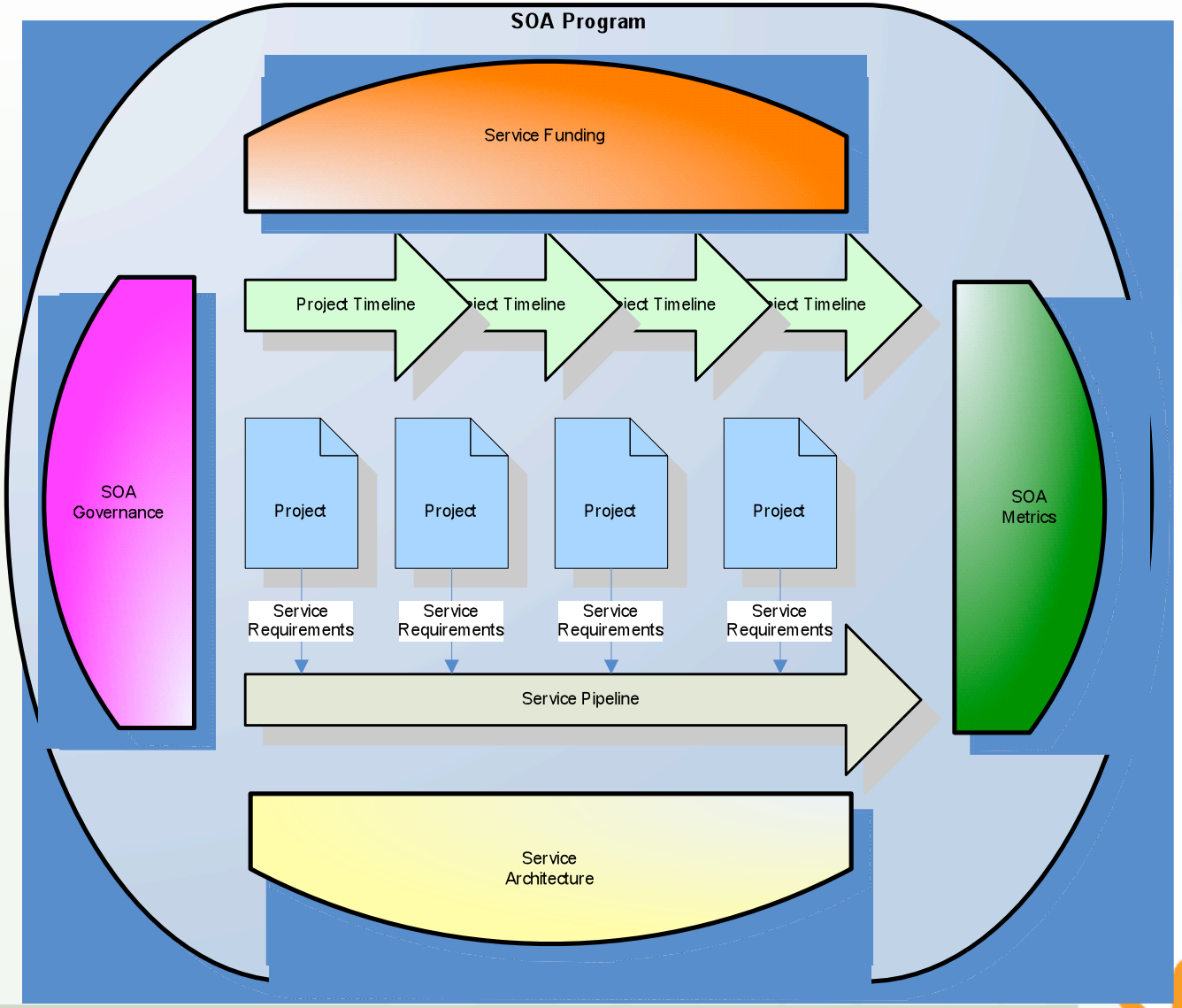
Setting up the SOA Lifecycle:

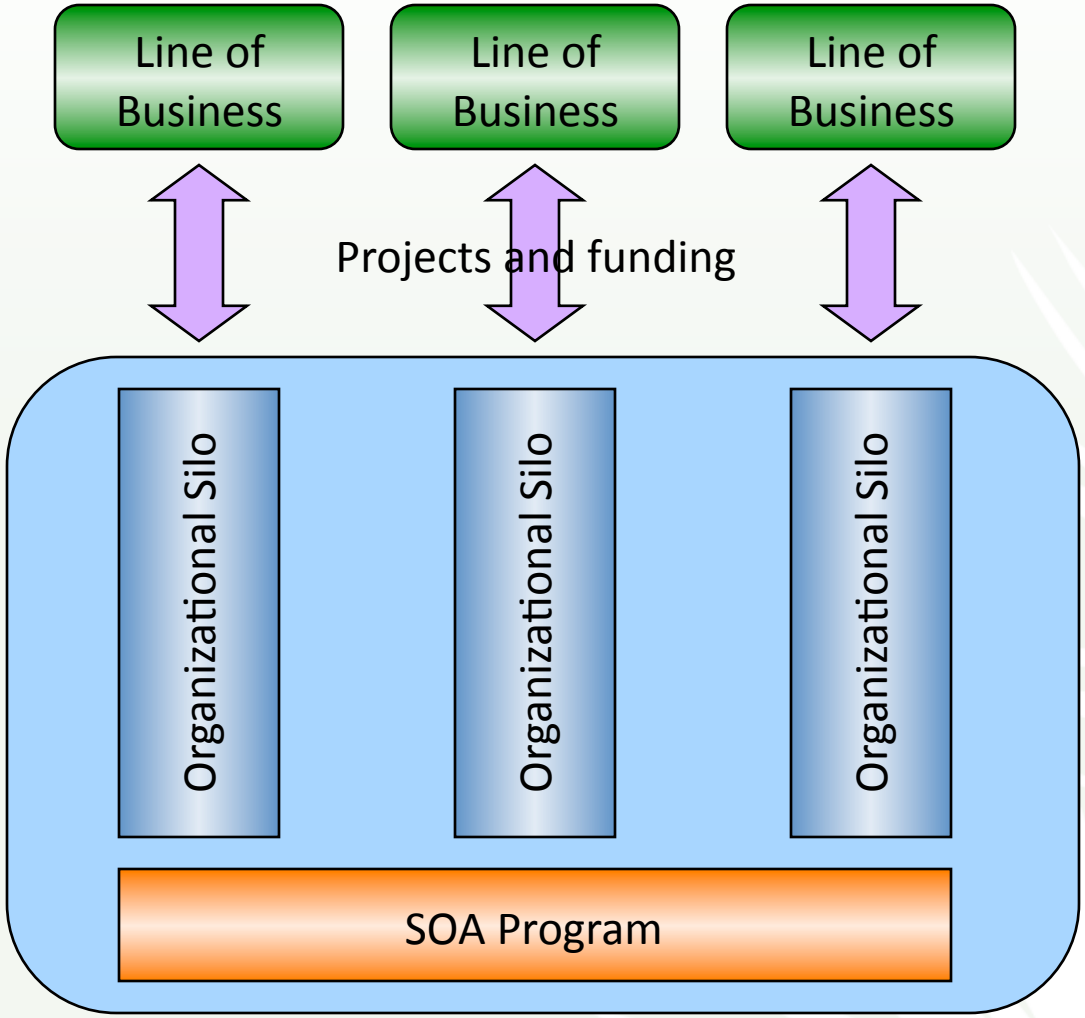
Delivering the SOA Vision, One Project at a Time

Leo Shuster
VP and General Manager of Enterprise Services
National City Bank (now PNC Bank)

Project-Oriented SOA Overview

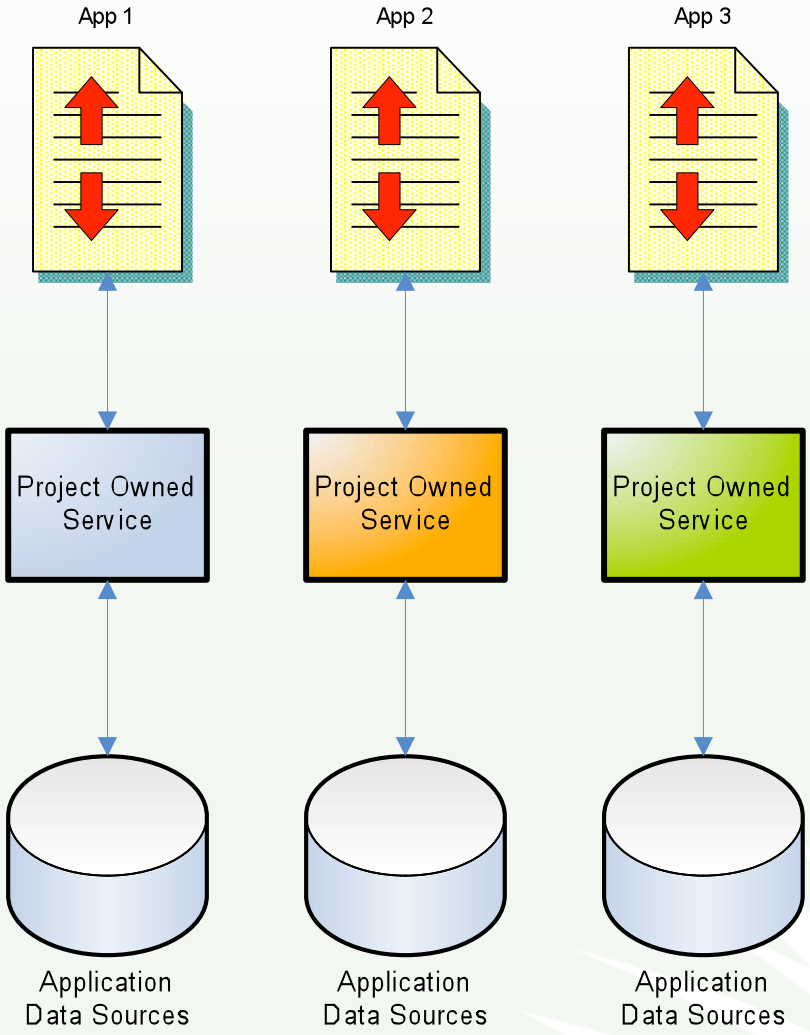


Organizational Reality



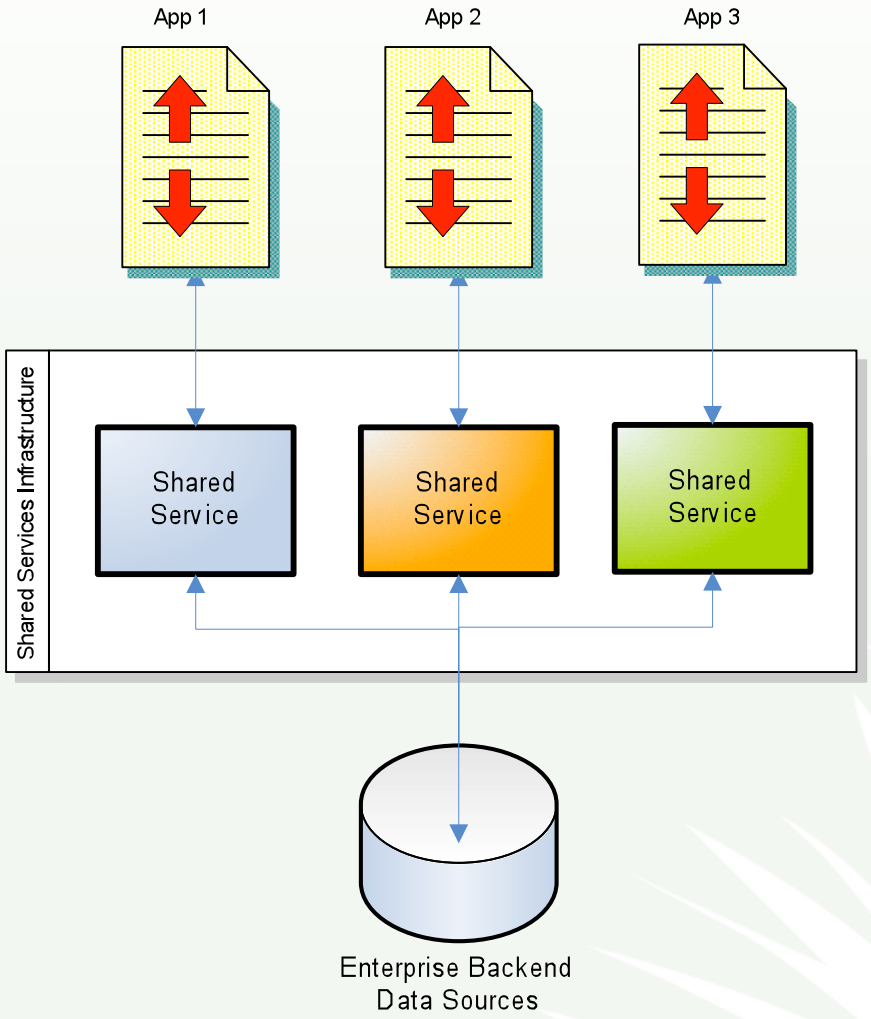
- **Projects are incompatible with SOA Programs**
- **Project attributes**
  - Time-bound
  - Focused on delivering specific outcomes to limited audiences
  - Concentrate only on their own requirements
  - Funding comes from a Line of Business
- **SOA Program attributes**
  - Span multiple groups and organizational silos
  - Goal is to establish reusable services for all consumers
  - Services have their own lifecycle
  - Must have central funding

Service Ownership Problem



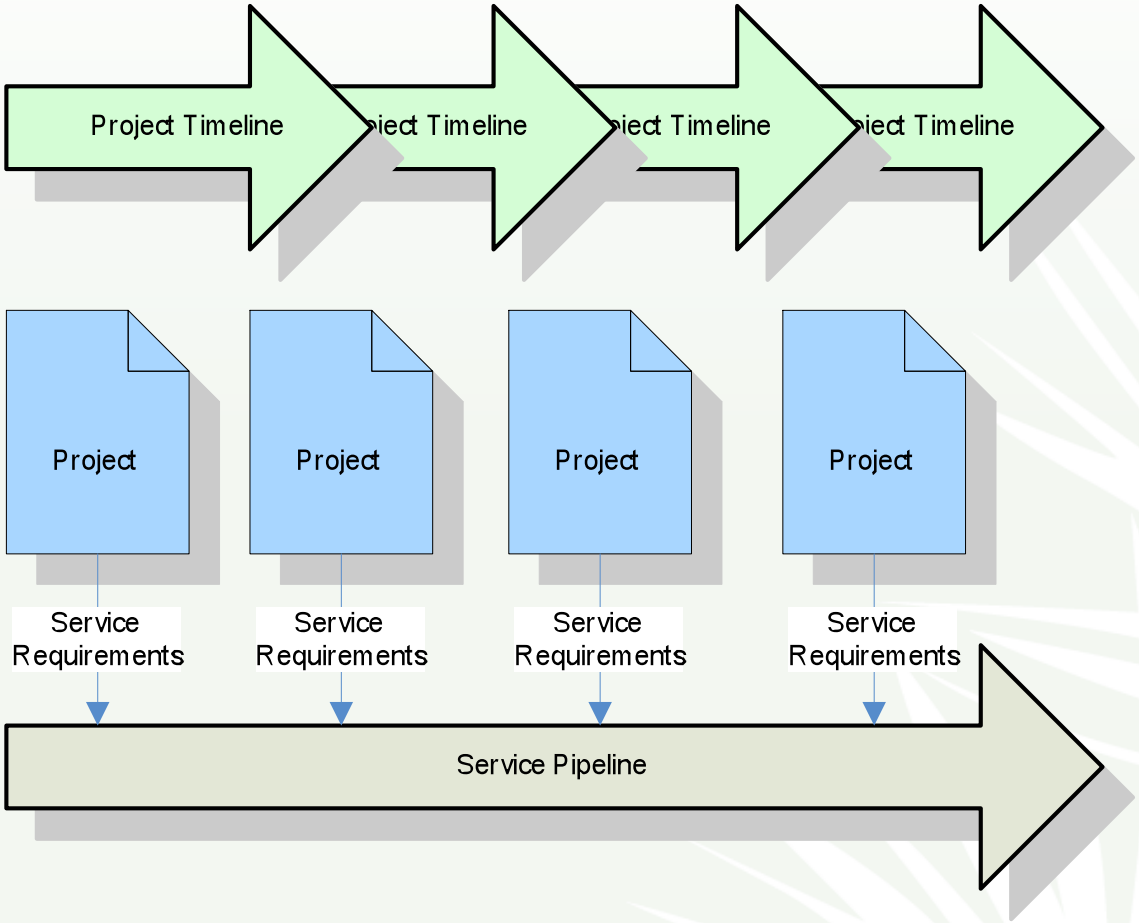
- **Project teams consider business logic their domain**
  - Consider themselves experts
  - Distrust others
  - NIH syndrome
- **Ownership**
  - Sensitive subject
  - Many IT managers have silo mentality
    - Assume they own the whole stack
    - Like to have control over every application component
    - Do not like to share control
    - Any perception of losing control can trigger an irrational response
  - Beware of empire building
- **Problem**
  - Projects are not structured to support shared services
  - Would have little incentive to address other projects' requirements

Service Ownership Problem



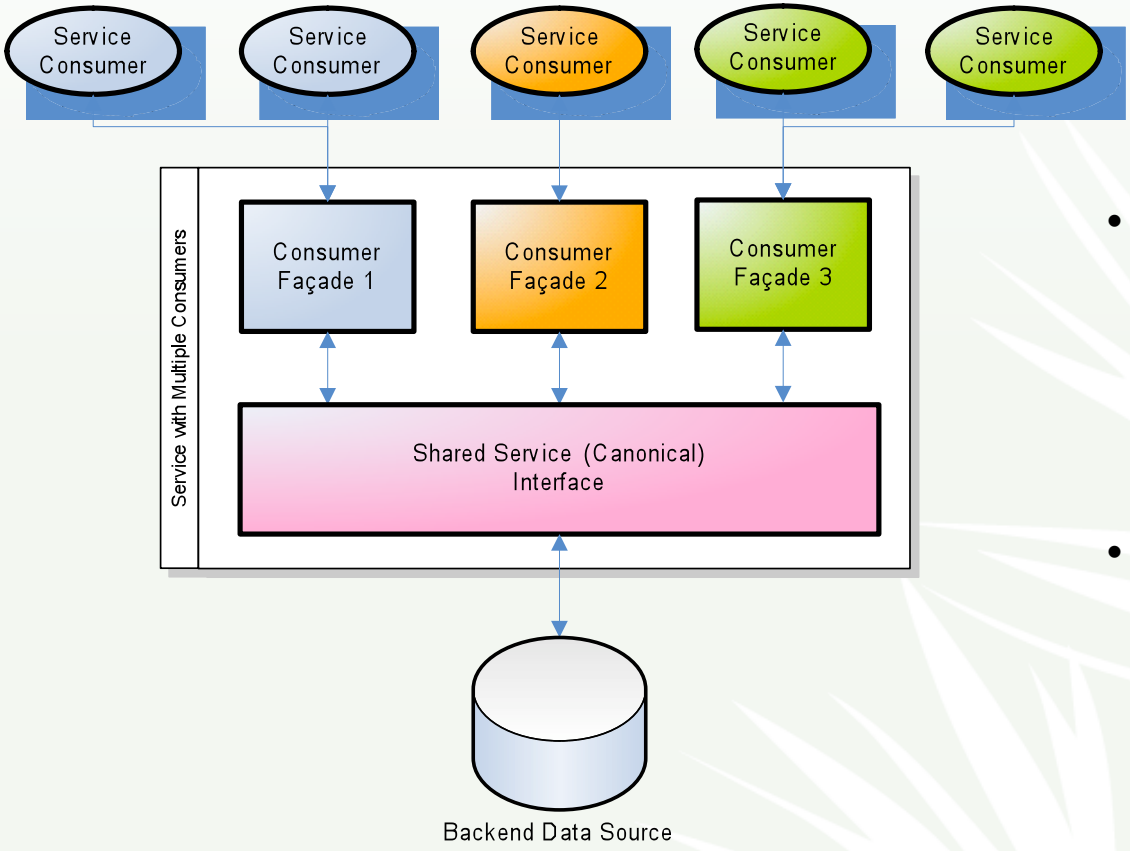
- **Services must be centrally managed**
  - Lifecycle different from that of a project
  - Code must be stored and versioned separately
  - Must reside on dedicated, independently scalable infrastructure
- **Ownership**
  - Enterprise shared services should be managed by a central team
  - Charged with reconciling all requirements and increasing service leverage
  - Central funding

Service Lifecycle Management



- **Service lifecycle**
  - Services created in response to project demand
  - New projects introduce additional requirements that need to be addressed
  - Services evolve independent of an individual project
- **Process**
  - Service lifecycle should be centrally managed
  - Central team should be charged with service identification, lifecycle management, and pipelining activities
  - All new requirements are incorporated into the services as they are discovered

Minimizing Impact of Changes and Maximizing Reuse

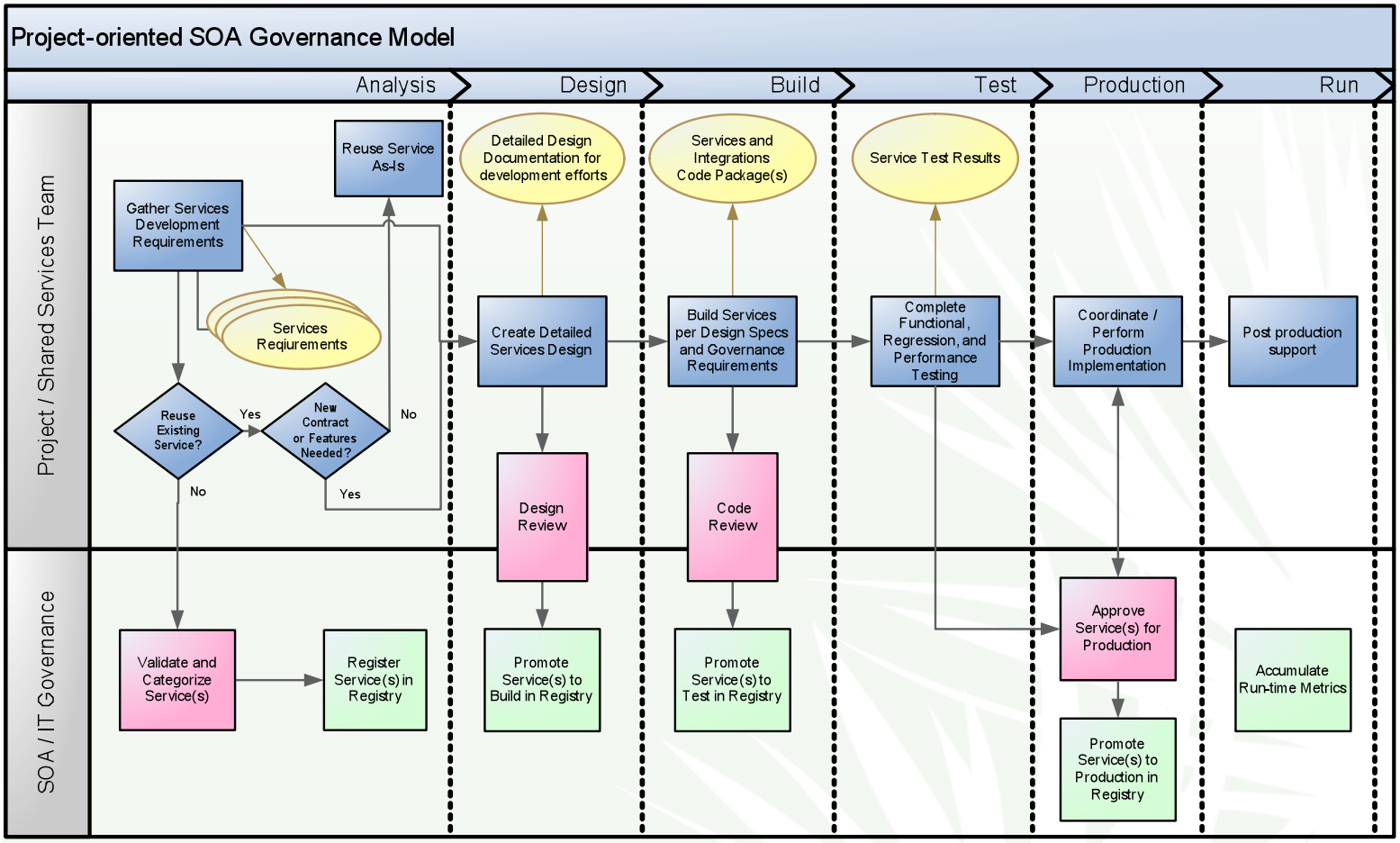


- **Changes are inevitable**
  - Services continue to change due to project demand
  - Service architecture must be flexible enough to accommodate changes
- **Canonical model**
  - Should be used to represent a consistent view of data
  - Reconciles differences between the same entities across the organization
  - Will change with service changes
- **Façade pattern**
  - Minimizes impact of internal service changes on consumers
  - Represents service contract specific to each consumer
  - Hides canonical model complexity

SOA Governance

- SOA Governance is critical to SOA program success
  - Maximizes service reuse
  - Encourages right behavior
- SOA Governance success factors
  - Align with internal Software Development Methodology
  - Minimize overhead
  - Maximize synergy with existing IT governance processes
  - Gain visibility of project pipeline as early as possible
  - Prefer influence over enforcement
- Process
  - Establish frequent governance checkpoints
  - Ensure project's compliance with previous recommendations and established best practices
  - Formal approval must be given before moving changes into Production

SOA Governance



SOA Funding

Funding Options

- Make the first project to build a service provide the complete funding
- Establish a central funding source that will cover all service design and construction expenses
- Provide supplementary funding to projects building services

Project- Based Funding

- Unfairly burdens the project
- Incompatible with SOA Program goals
- Will result in project-owned or hard to reuse services

Supplementary Funding

- Most pragmatic
- Central fund established and made available to projects
- Centrally managed
- Covers costs outside of project scope

Central Funding

- The easiest approach
- Hard to convince IT management
- Presents opportunities for abuse
- Strong governance needed
- May require a chargeback mechanism to be established

SOA Metrics

- Metrics are needed to:
  - Measure SOA Program effectiveness and level of adoption
  - Communicate results
  - Meet established goals
- Steps to capture appropriate metrics
  - Capture all the services being created
  - When completed, determine the cost to build each service
  - Determine integration costs related to service reuse
  - Capture all reuse opportunities

Most Popular SOA Metrics

- **# of services created**
- **Amount of service reuse**
- **Cost avoidance/savings**
- **Projects using services**
- **ROI**

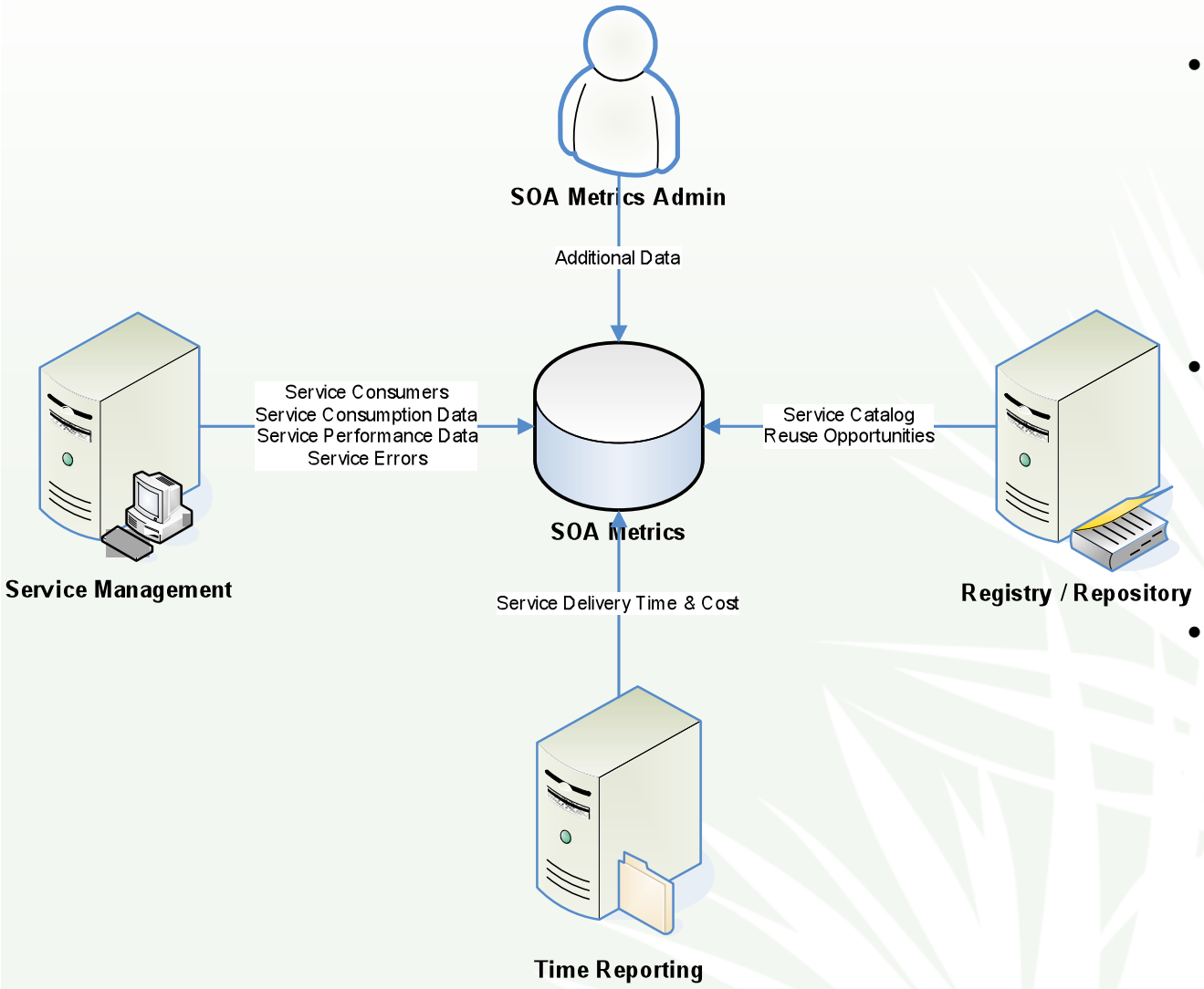
Calculating Service Cost Avoidance

Service Cost Avoidance = Service Build Cost – Project's Service Integration Cost

Where

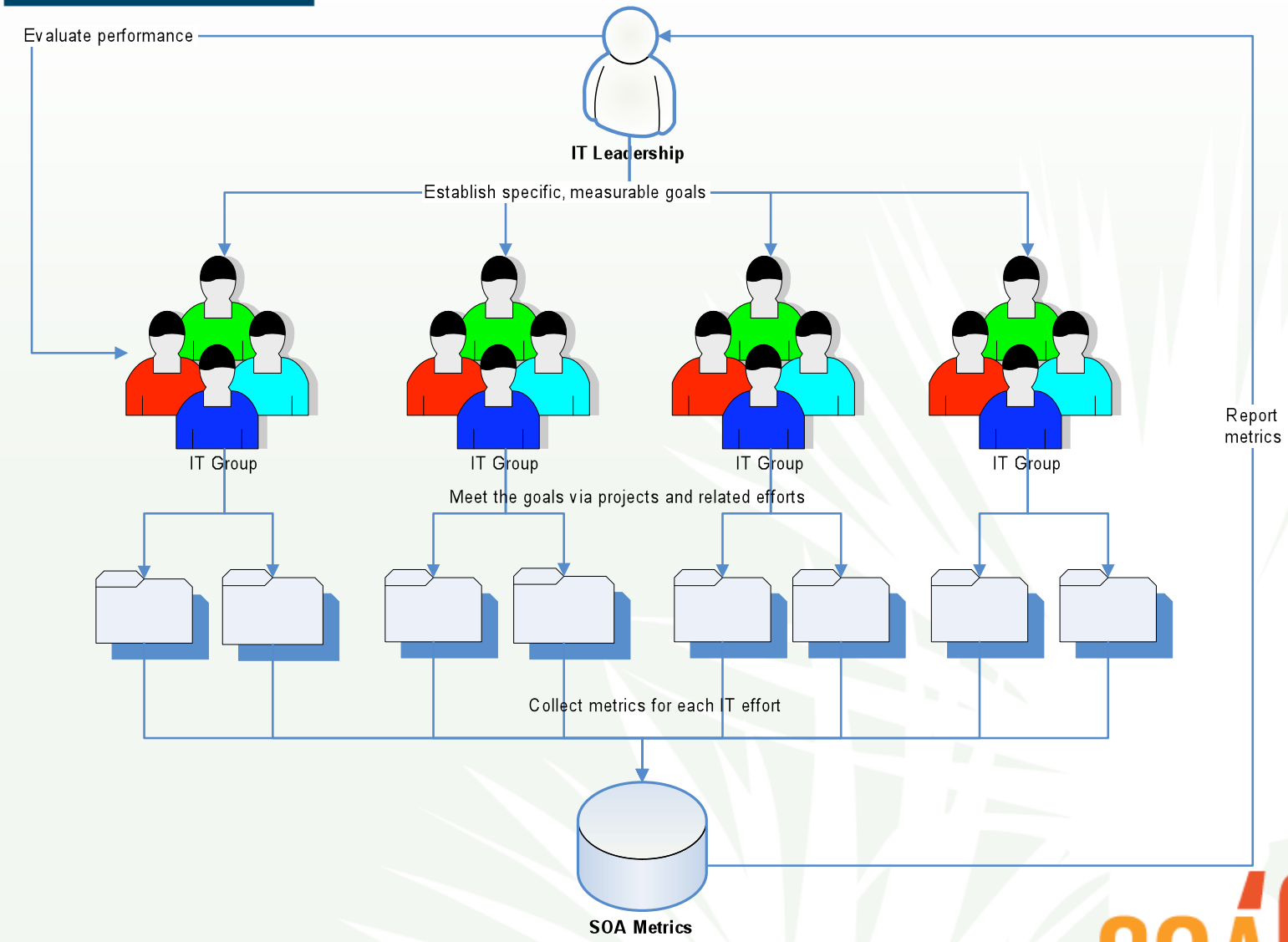
Service Build Cost = Initial Service Build Cost + Cost of all Subsequent Changes

Collecting SOA Metrics

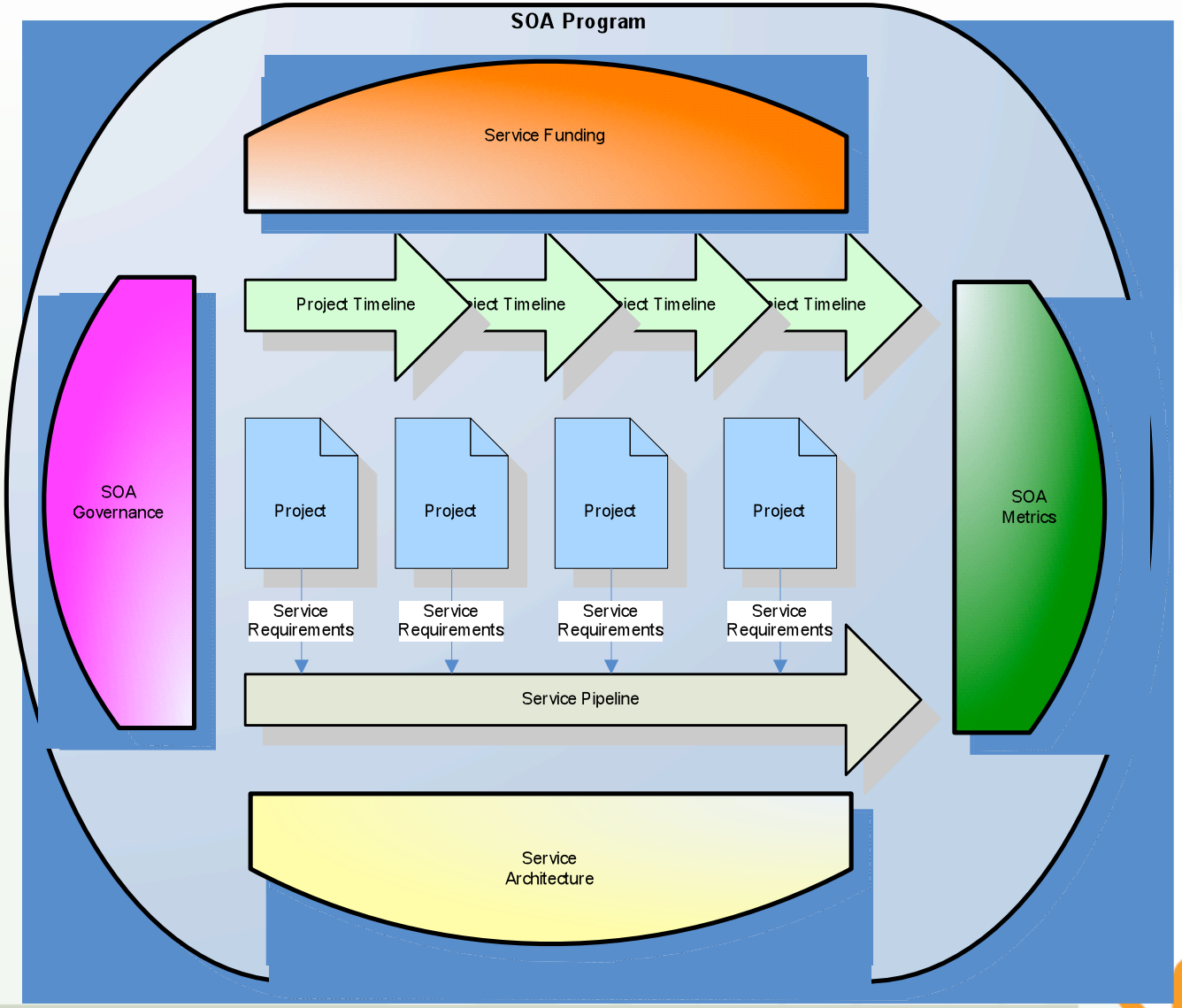


- **Automation is key**
  - Eliminates errors
  - Ensures consistency of data
  - Reduces manual data entry and administration efforts
- **Leverage existing technology**
  - Many products will help in metrics collection
  - Extract data directly if possible
- **Time tracking is tricky**
  - Need to track design, development and testing time by service
  - Time reporting software should be set up appropriately

Improving Service Reuse



Delivering SOA Vision



References

- *Making SOA ROI Real*: http://soa.sys-con.com/node/847118
- *Project-oriented SOA:* http://www.soamag.com/I21/0808-2.asp
- Leo's Blog: http://leoshuster.blogspot.com/
- E-mail: leo.shuster@nationalcity.com