

Application Architecture in a Distributed World

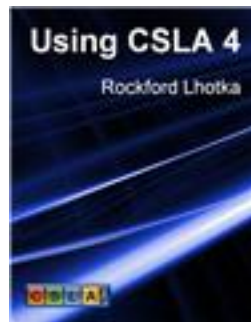
Rockford Lhotka
Magenic
rocky@lhotka.net

Magenic

Custom solutions that fit. Guaranteed.



Microsoft
Regional Director
PROGRAM



Architecture

Principles/structures that transcend apps

- Not design!
- Not per-app!

Horizontal concerns

- Logging
- Security

Vertical concerns

- UI <-> Biz Logic <-> Data
- Cross-layer/cross-tier communication

Layered Architecture

Organize
your...

Application code

Test plans and execution

Development team

Manage
complexity

Related functions go in a layer

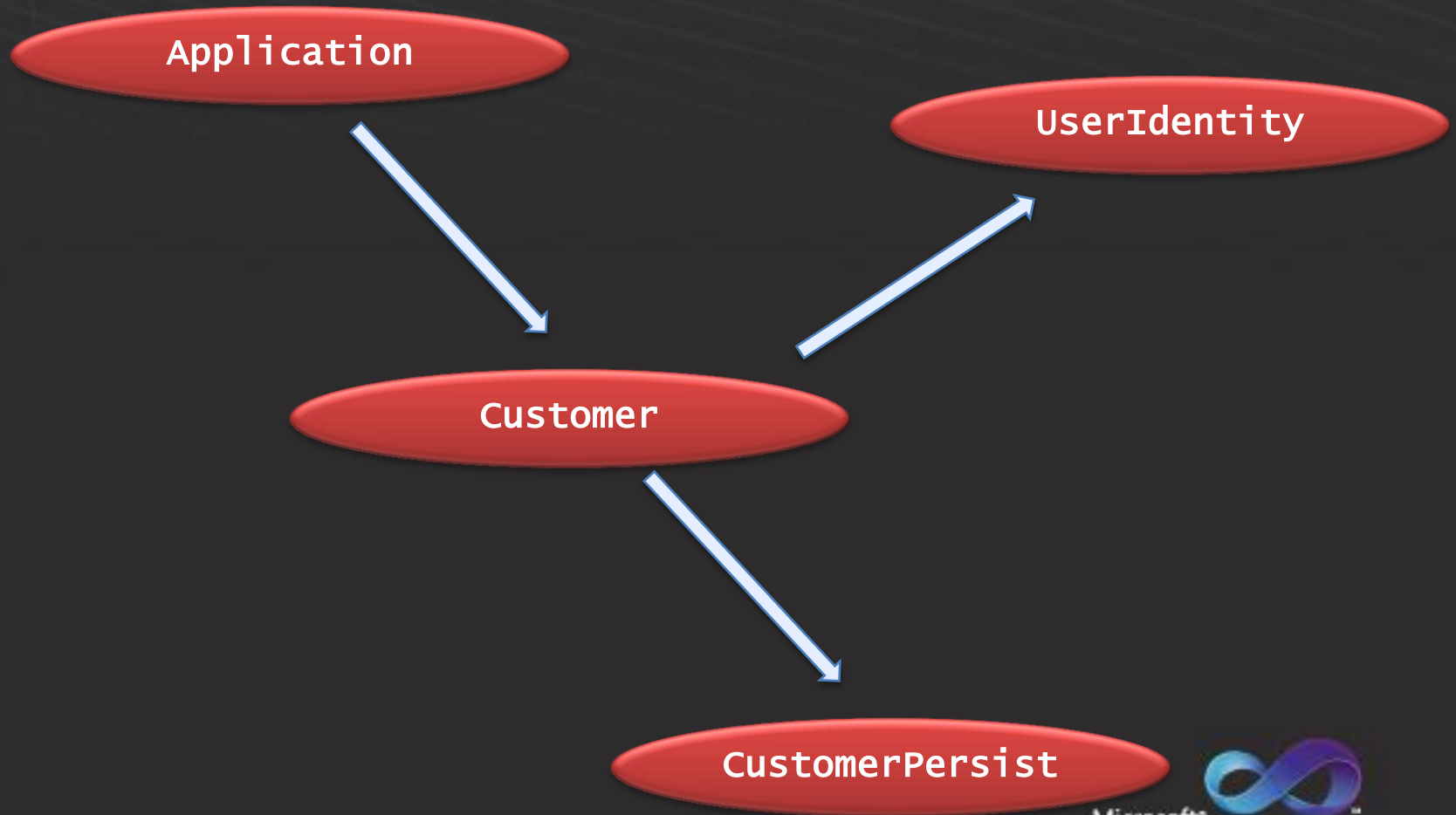
Layers are isolated

Layers communicate

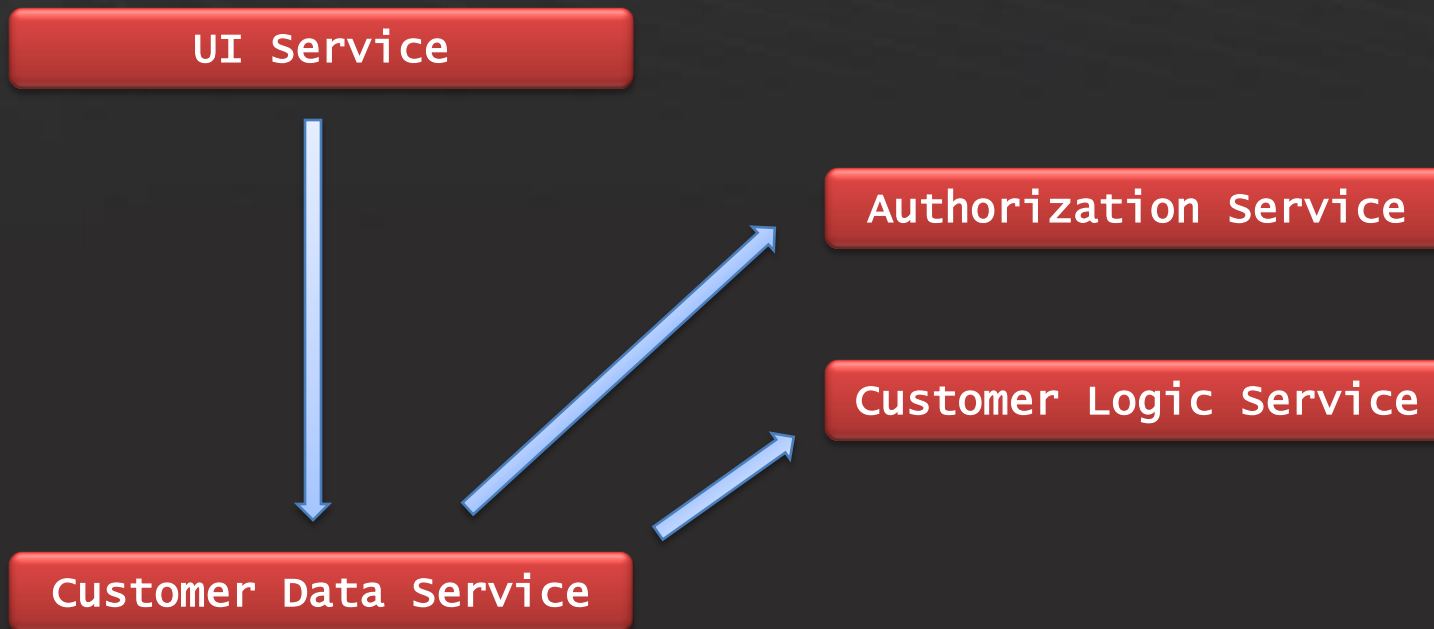
Layering Software: Procedures



Layering Software: Components



Software Layering: Services



Software and Physics

- **Law of Conservation of Energy**
 - The total energy in an isolated system is constant despite internal changes

- **Law of Conservation of Logic**
 - The total logic in an isolated system is constant despite internal changes

Today's Options

N-Tier

- Good for single app
- Fast, efficient, relatively low cost

SOA

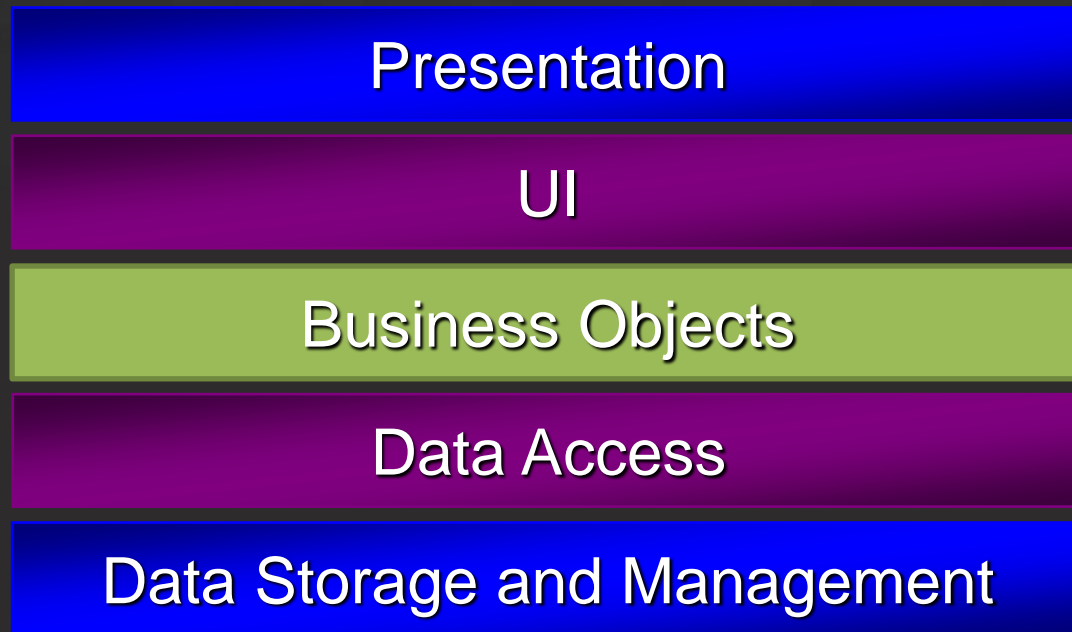
- Good for system of many apps
- Open standards, loosely coupled

Application Layering: Logical Layers

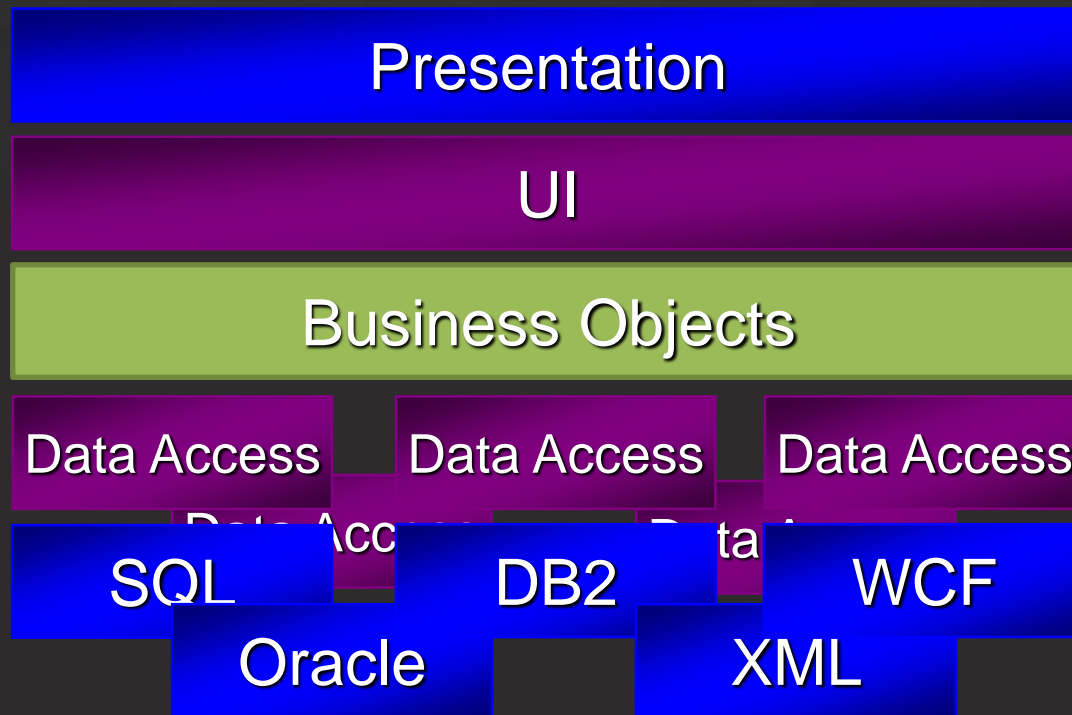
- Breaks application into logical layers
 - Layers may all run on machine
 - Layers may run on separate machines
- Provides
 - Maintainability
 - Readability
 - Flexibility
 - Reuse
 - Lower development/maintenance costs



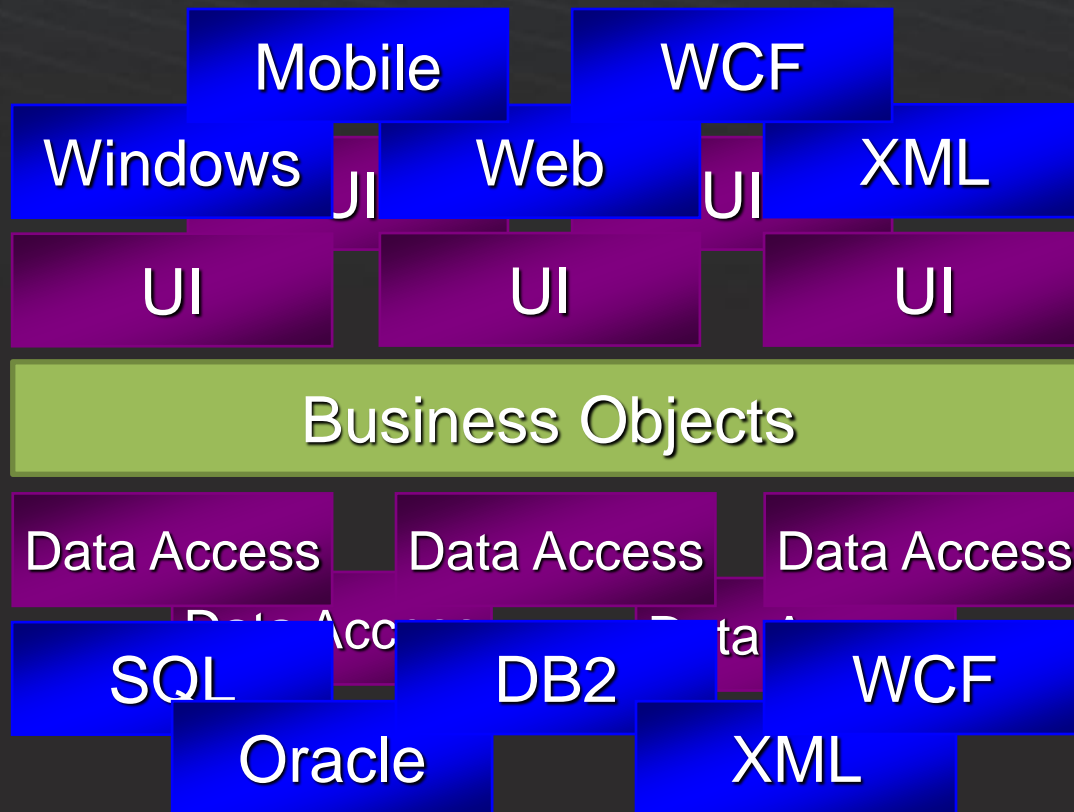
N-Layer



Multiple Data Sources



Multiple Presentations



Physical Tiers

- Deploy logical layers to physical tiers
- Trade off between
 - Performance
 - Scalability
 - Fault tolerance
 - Security

Historical Perspective

Single thread,
single memory

- Traditional; Win16,
OpenVMS

Multiple thread,
shared memory

- Parallel; Win32,
Unix, Amiga

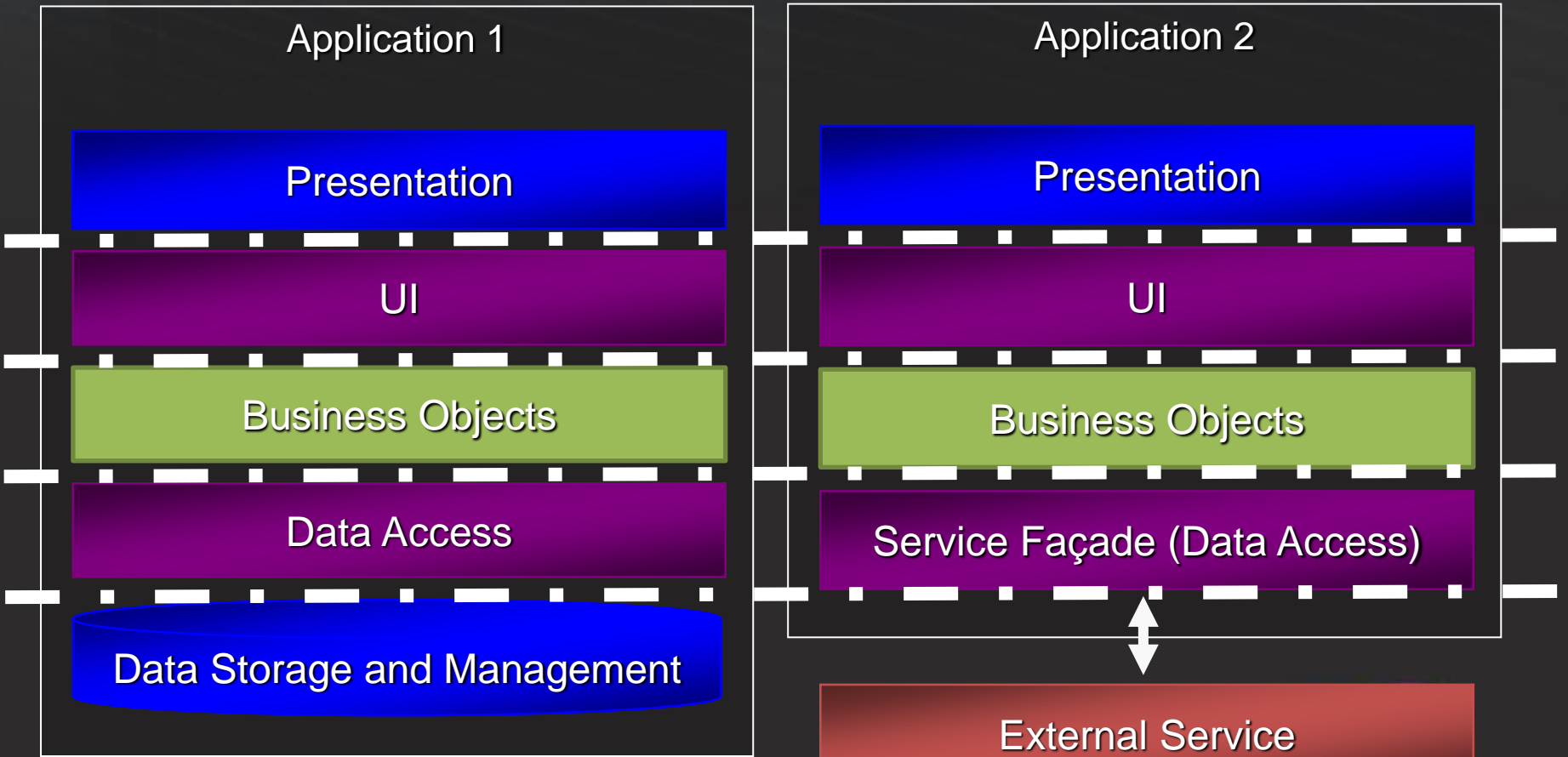
Multiple thread,
multiple
memory

- Distributed; COM,
n-tier

Async multi-
thread, multi-
memory

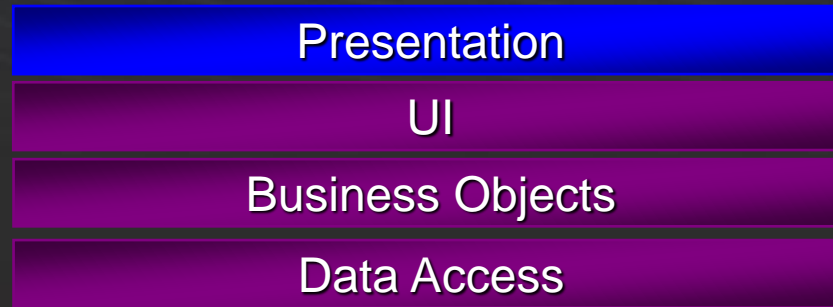
- Distributed parallel;
Grids, SOA

N-Tier

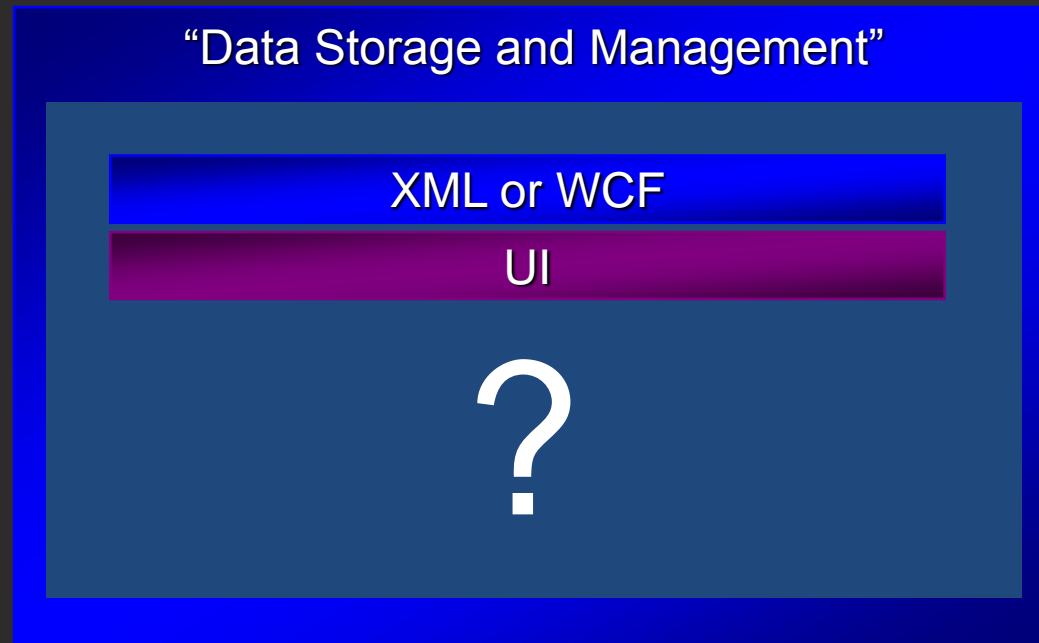


Service Oriented System

Client
Application

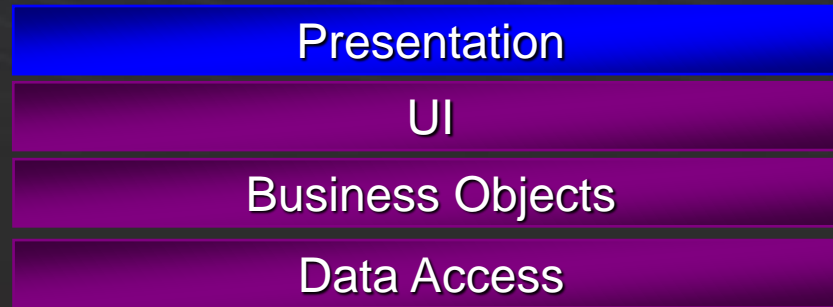


Service
Application

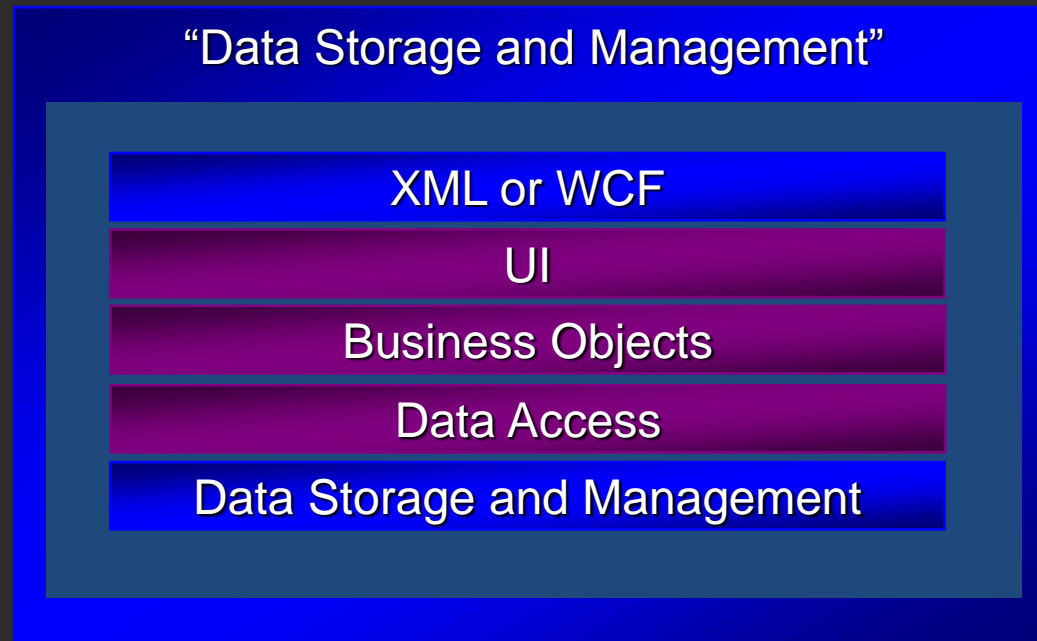


Service Oriented System

Client
Application

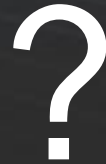


Service
Application



Service Oriented System

Client
Application



Data Access

“Data Storage and Management”

XML or WCF

UI

Business Objects

Data Access

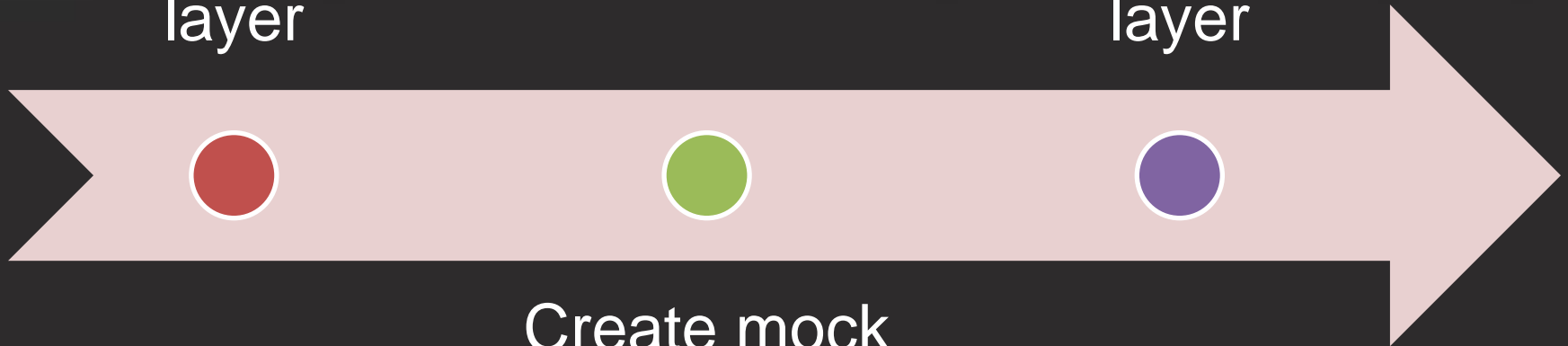
Data Storage and Management

Service
Application

Test Layering

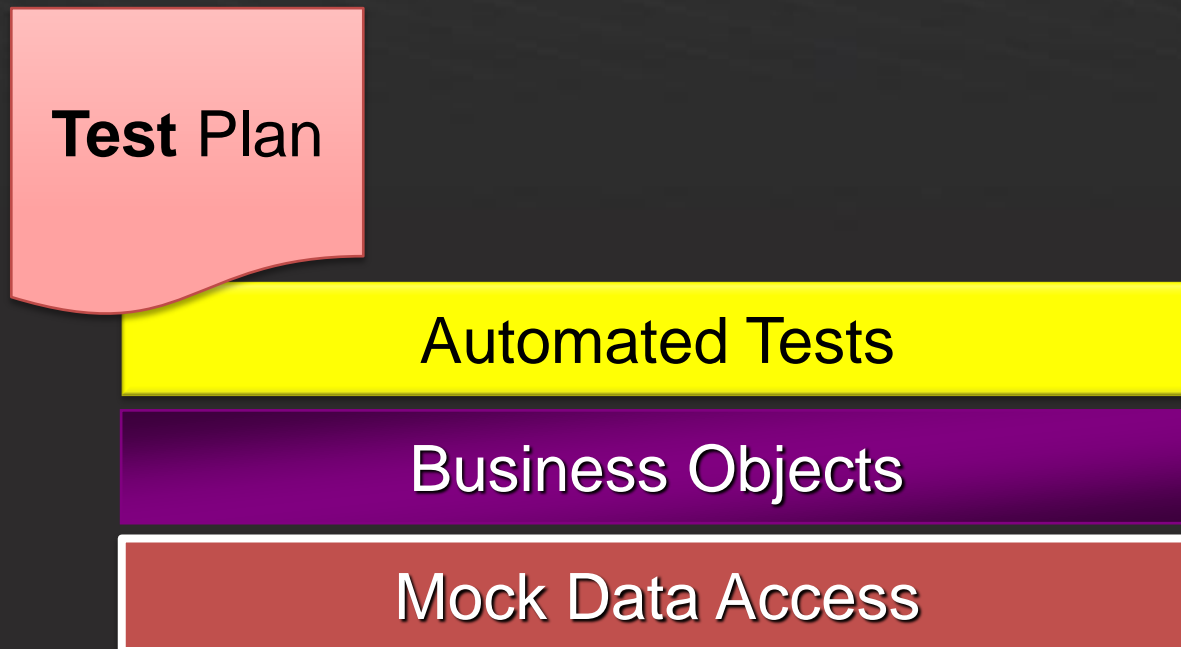
Create test
plans for the
layer

Write tests
against the
layer



Create mock
objects for
next lower
layer

Layered Test Example



Layered Testing

Works well with

- Business layer
- Web services/WCF interface
- WPF interface

Workable with

- Data layer (using database dependency)
- Web interface (can often automate tests)

Difficult with

- Windows Forms (hard to automate tests)
- Silverlight
- Windows Phone

Layered Teams

By technology

- Presentation
- Business layer
- Data access
- SQL

By business knowledge

- Customer
- Product
- Inventory
- Shipping

Frameworks

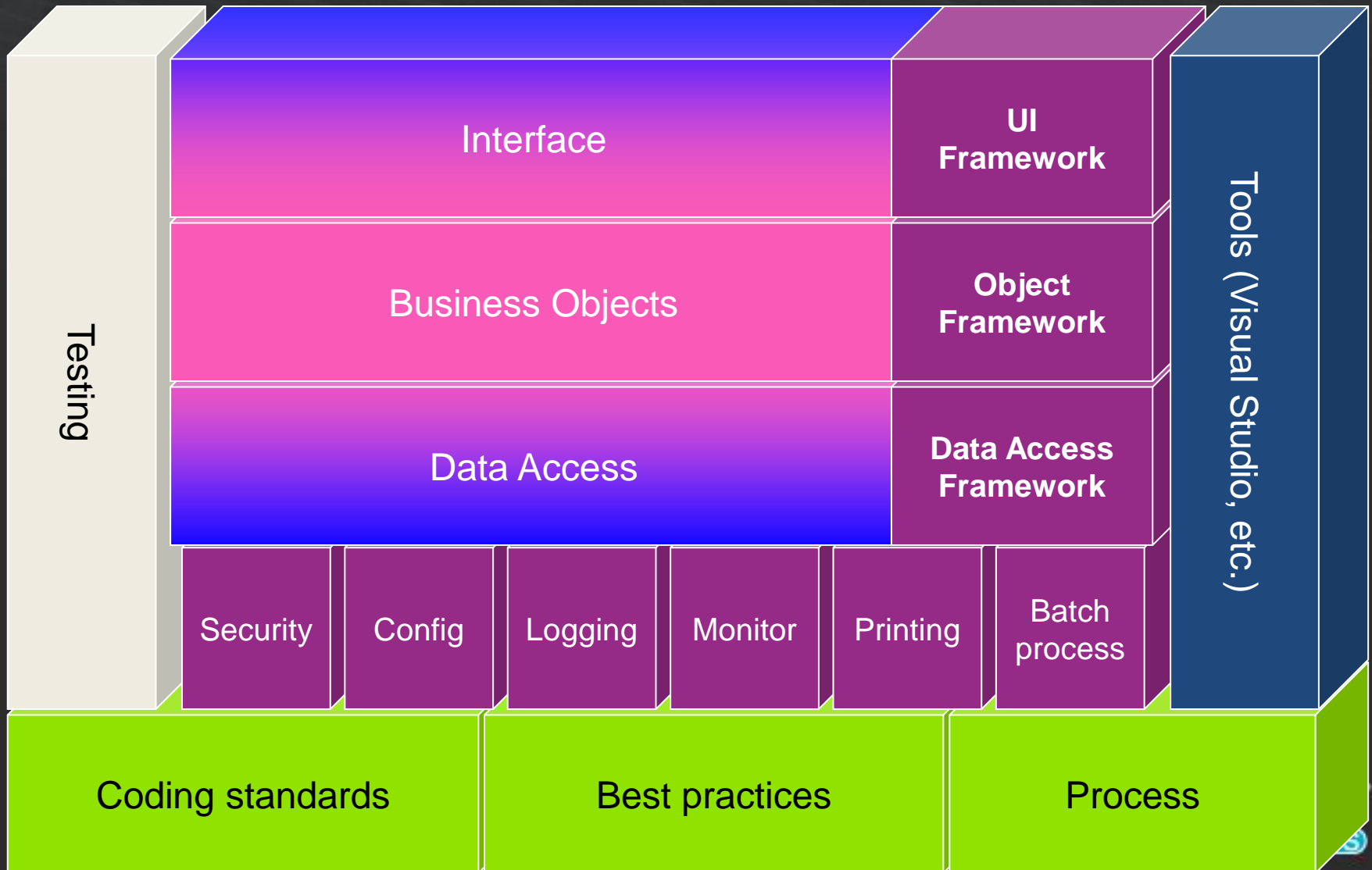
Get devs to do the “right thing”?

- Make the right thing the easy thing

Codification of architecture

- Framework implements “plumbing”
- Framework + Tools = Productivity

Framework Areas



Summary

All apps should be n-layer

- Organize code to increase maintainability

Most apps are n-tier

- At least 2-tier, often 3- or 4-tier

SOA systems necessary

- Loose coupling between apps *required*

Your Feedback is Important

Please fill out a session evaluation form
drop it off at the conference registration
desk.

Thank you!

Magenic

Custom solutions that fit. Guaranteed.



Microsoft
Regional Director
PROGRAM

