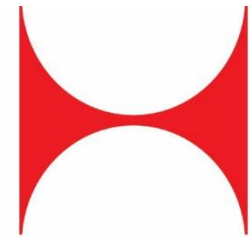


Pulling Back the Curtain on Application Modernization

PUG Challenge 2016

Tim Kuehn

Senior Consultant



TDK
CONSULTING

www.tdkcs.com

Inspired by CEOs that've asked "Why should I be concerned? My customers aren't asking for this."

Something for everyone

Expose the audience to new possibilities

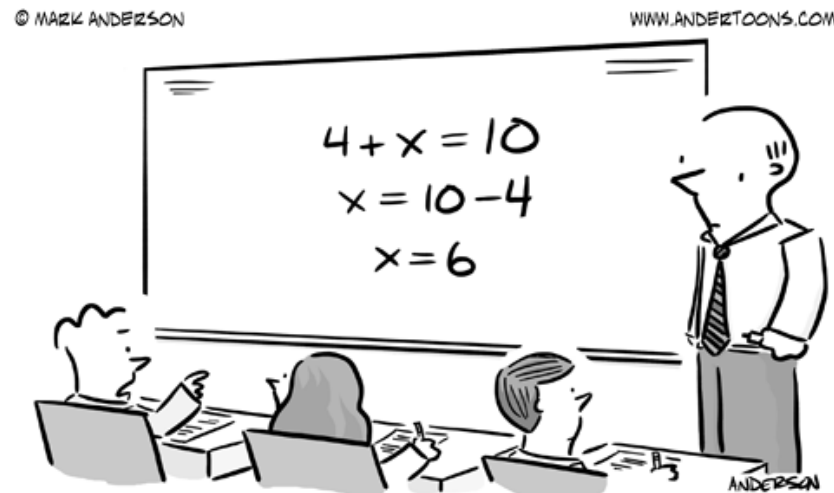
What is Modernization?

Modernization is needed for applications written by competent and ingenious developers that created innovative solutions to business problems of the time....

...that have been eclipsed by advances in business and technology.

Modernization has been going on for decades – such as the great migration from mainframes and minis to PCs...

...and goes beyond technology to include modernized thinking



"Hold on. When we learned Roman numerals, X was 10. Now it's 6. What's going on around here?!"

What Changed?

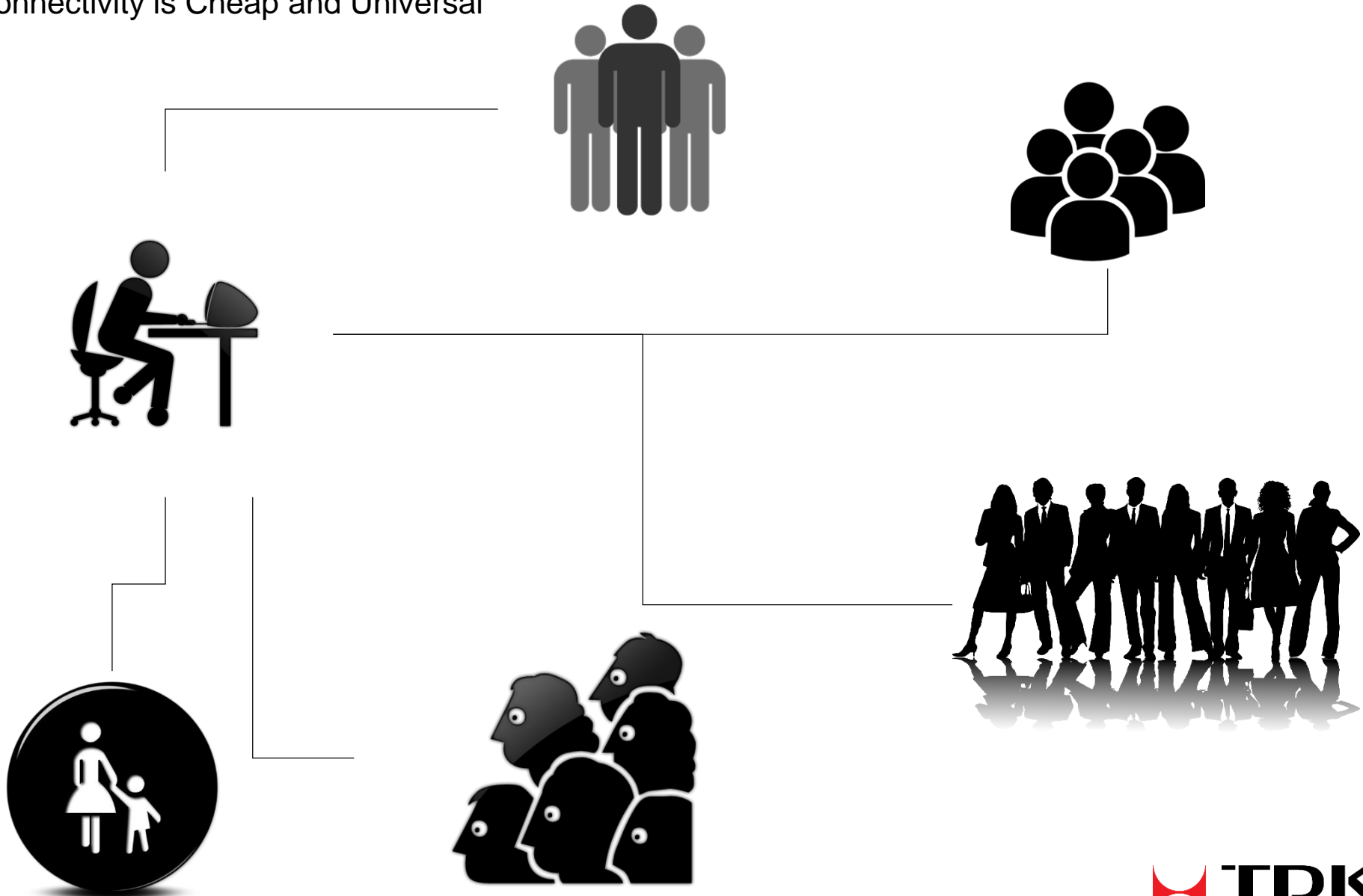
Old	New
Self-Serve, Client / Server	Appserver, Messaging, WSA, REST
Include-base Data Browsers	Create BROWSE, UI Frameworks
BL in include files, super procedures, persistent procedures	OO, Classes, Java, Javascript, Scala, Haskell
UI / BL / DA entangled	UI / BL / DA separation, OERA, OEAA
ChUI, Progress GUI	.NET, Kendo UI, Xamarin, Angular UI
Windows, Linux / UNIX, Mainframe	Windows, Linux / Unix, iOS, Android, BB, IOT
Single Server – Multiple Applications	Single Physical Server – Multiple Virtual Servers, Cloud Computing, Cloud Services, Microservices, Docker, Lambda
Progress, SQL, Relational Databases	Progress, SQL, Graph, Eventually Consistent, Document Databases
Hand Coded Reports, Report Writers	Big Data, Analytics, Machine Learning (Google Tensor)

Communications Speed

Acoustic Coupler	110 Baud 300 Baud
Modem	1200 Baud 2400 Baud 56K Baud
Broadband	Cable, DSL Fiber Cable v2, DSL v2

What Changed?

Connectivity is Cheap and Universal



It's a Matter of Survival

- ChUI / GUI version harder to sell
- Need an updated user interface
- Need updated functionality / connectivity requirements
- Younger workforce expects mobile / browser look & feel

Disruptive forces

What's causing this disruption? IBM's recent [Global C-suite Study](#) shows that executives believe *technology* and *market factors* are the *biggest external forces transforming the competitive landscape in many industries*. It also states that regulatory concerns, macro-economic factors, and people skills are playing a role in the changing business environment.

The warning signs of digital disruption

How do you know when your industry is at risk of digital disruption?

Uber's success in the taxi business – and widespread speculation that it will have a similar effect on the delivery and logistics sectors – has made many other industries wonder if they could be next.

Here are some signs that your business might be ripe for disruption.

- 1. Venture capitalists are circling** - If lots of venture investment is going into companies that sound like they might be part of your industry -- but aren't exactly -- then watch out. This means that outsiders see opportunity in your sector, and are coming at it from new angles.
- 2. You're using technology to cut costs, rather than create value** - Customers understand that investments in technology allow companies to decrease overhead and personnel, and generally to do business more cheaply. But an industry that uses technology only for these purposes, and doesn't leverage it to improve customer experiences, is in danger of blindsiding.
- 3. Your customers are getting older** - Perhaps your product or service works just fine for those who are already comfortable with it. But if your customers have steadily been getting older, the chances are that your industry is not delivering in a way that appeals to Millennials. Someone who figures out a more user-friendly way to deliver the same offering will win the Millennial market. Over time, older customers will appreciate the ease-of-use, and gravitate to the upstart, too.

- 4. Your service is very high-touch** - If you've got a lot of customers who spend a lot of time getting guidance from your staff, chances are that *much of that interaction could be automated*. There may even be a new group of customers – again, Millennials – that would *prefer to get their information digitally rather than in a one-on-one human interaction*.

In the financial services industry, for example, the idea of a "robo-advisor" to help manage one's finances initially sounded like a joke a decade ago. Now so-called *robo-advisors represent a legitimate threat to the financial services establishment*. (And teaching....)

- 5. Your customer satisfaction is low** - If customers are sticking with you mostly because they don't have many choices, you should be worried. You may have been temporarily protected by barriers to entry, such as complex regulations or the necessity of making massive investments in order to truly compete. As Uber and AirBnB have shown, technology can trump these – quickly.

Is Your Business at Risk?

Disruptors such as Uber prey on complacency – on industries that haven't bothered to innovate, or to embrace technology, often because it simply seemed like too much trouble and because their customers had few other choices. But as the **cost of technology drops** and **its effectiveness increases**, *the barriers that had shielded some of these sectors from dramatic change are increasingly being breached.*

If that sounds like your business, it's time to innovate – or watch an outsider do it for you.

<http://www-935.ibm.com/services/c-suite/study/perspectives/is-your-company-about-to-be-uberized/>

Businesses that use customer lock-in as an "economic moat" are the most at risk.

The term economic moat ... refers to a business' ability to maintain competitive advantages over its competitors in order to protect its long-term profits and market share from competing firms. <http://www.investopedia.com/ask/answers/05/economicmoat.asp>

What are the Disruptors Doing?

1. The world's largest taxi company (Uber) owns no taxis
2. The world's largest accommodation provider (Airbnb) owns no real estate
3. The largest communications companies (Skype, WhatsApp, Facebook Messenger, Viber) own no infrastructure
4. The world's most valuable retailer (Alibaba) has no inventory
5. The most popular media platform (Facebook) creates no content
6. The fastest-growing banks actually have no money
7. The world's largest movie house (Netflix) owns no cinemas
8. The largest software vendors don't write the apps (Apple, Google, Facebook)



Each of these companies drive efficiencies by connecting producers and consumers without regard to location.

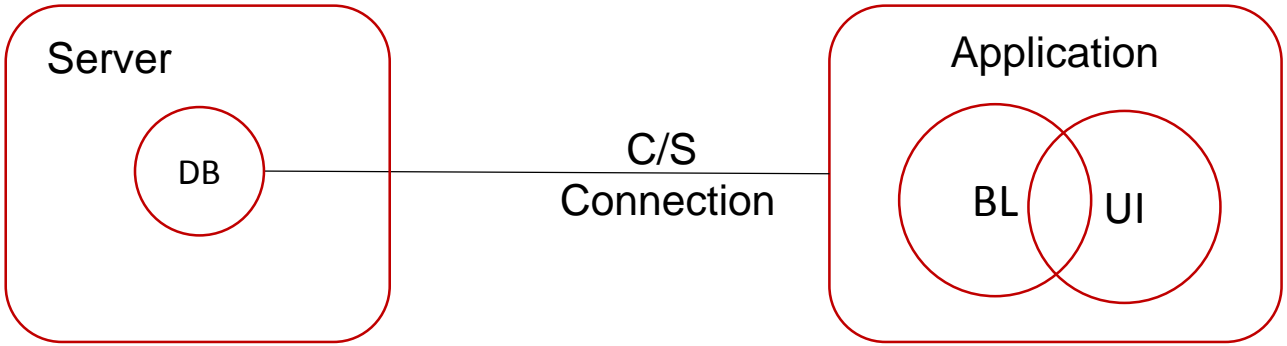
This eliminates a significant transaction cost – efficiently finding a counterparty to conduct a transaction with.

It also creates an opportunity for builders of better mousetraps to eat the Disruptors eggs. Anyone remember the pre-google search engines?

How Should Business Respond?

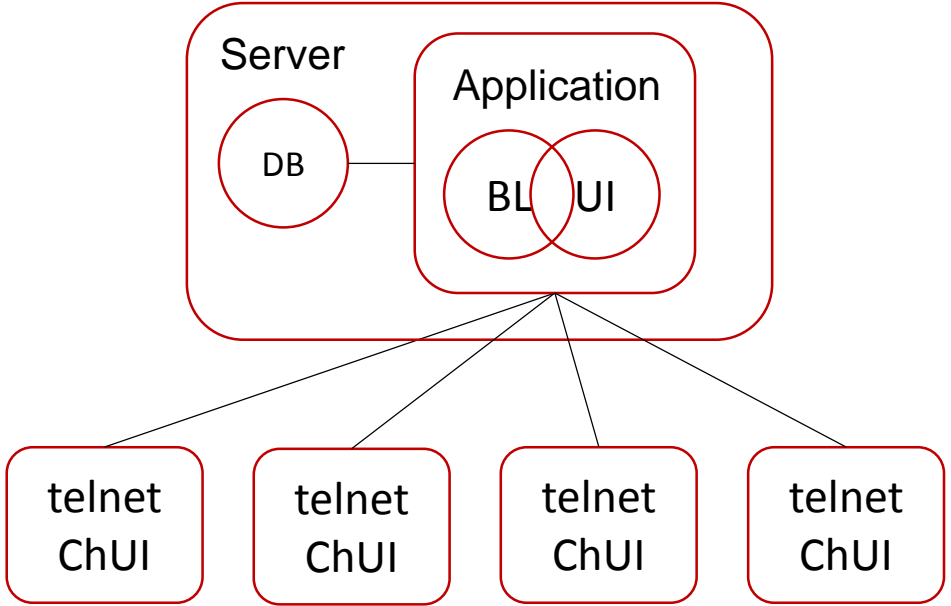
Business applications need a more connectable architecture with a view to attaining maximum business agility

Desktop C/S Application

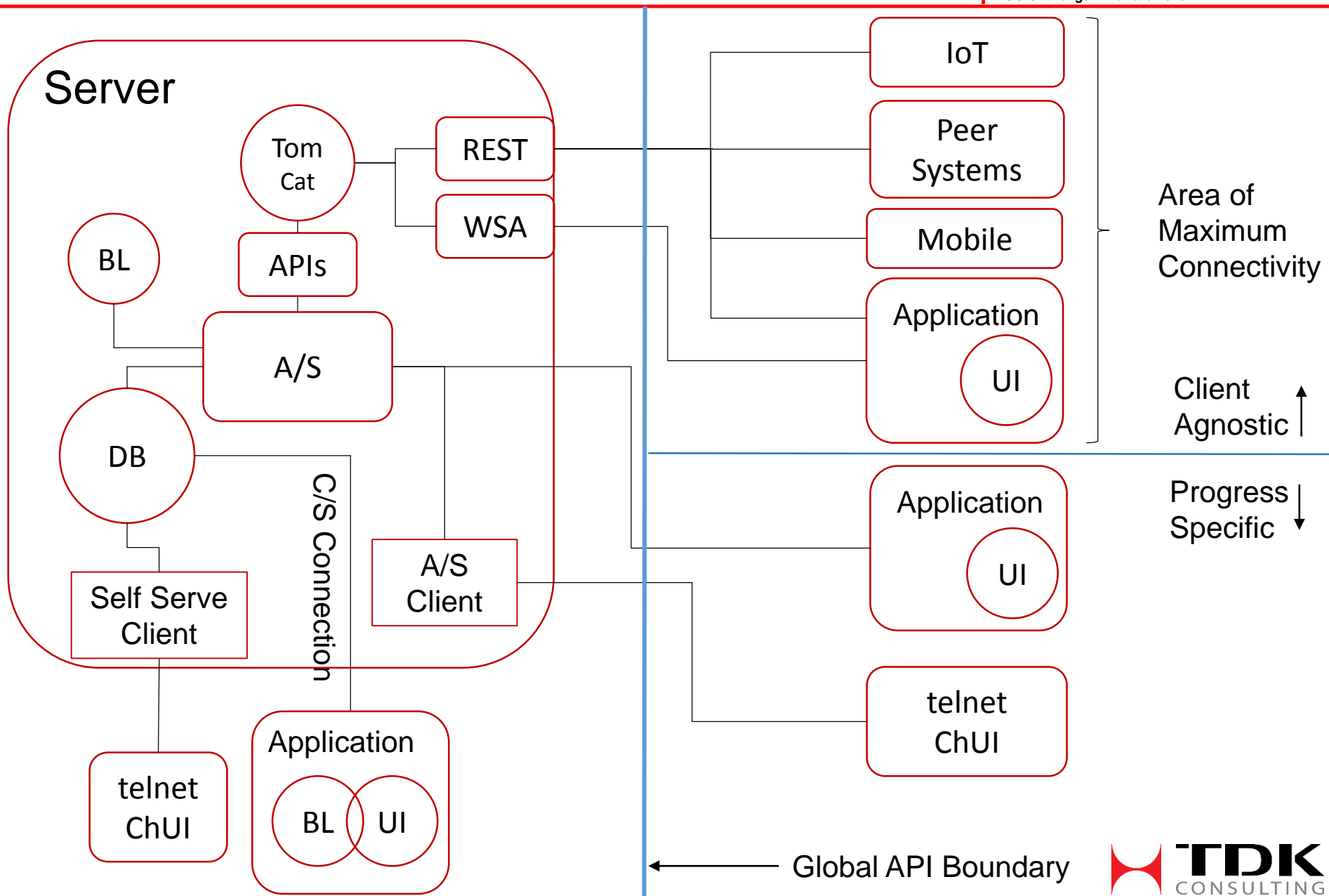


- ChUI
- Progress GUI

Self-Serve ChUI



Choose Your Interface



Separation of Concerns:

- Avoids locating multiple kinds of functionality in the same block code.
- Related to Single Responsibility Principle at the lower layers (ie do one thing and one thing only)
- Implements the “Don’t Repeat Yourself” Principle.
- Supports Decoupled Architectural layers of functionality (OERA, OEAA)
- Results in more maintainable systems that can be readily adapted to new uses
- Can be a challenge to recognize mixing of concerns when first writing the program

Decoupled Architecture:

- A decoupled system consists of multiple, stand-alone modules that are brought together to perform a task
- Each module doesn’t know about the others
- Overall activity of the collection is determined by an integration unit that ties the module functionality together

Decoupled Modules:

- Components of a system which have well-defined APIs and only communicate through those APIs
- Enables the independent development of different components/layers without having to wait for the code on the other side of the API to be complete.
- Gains flexibility at the cost of simplicity
- UI / BL separation is a classic example of decoupling

The End Goal Is: Agility

The path to get there is -

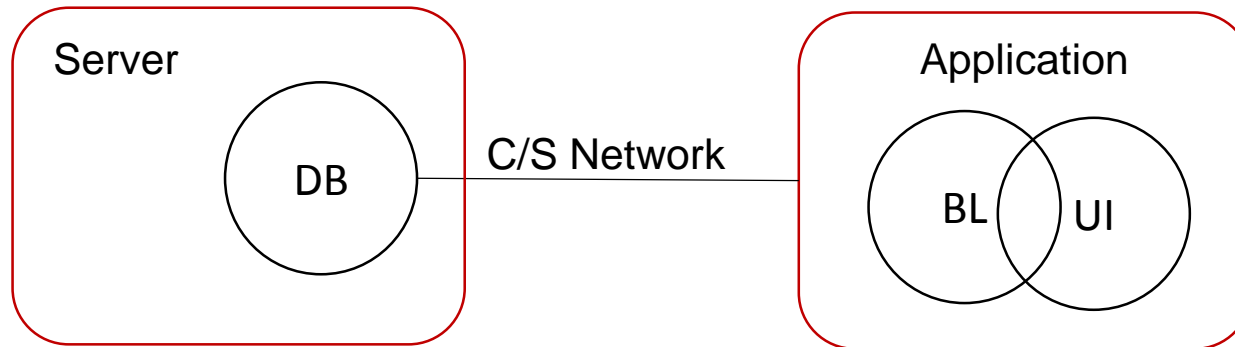
- Good Separation of Concerns
- A Decoupled Architecture
- Good API Protocols
- Matching the Right Technology to the Requirements

The benefits are:

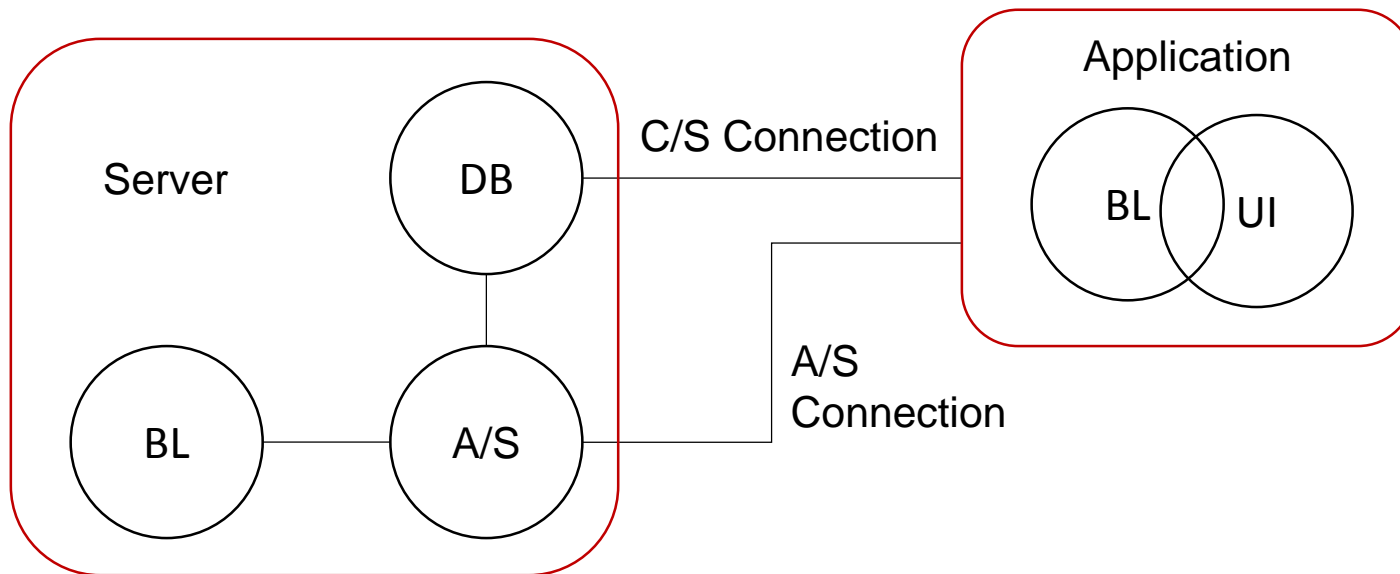
- Decreased time to market
- Reduction in Technical Debt
- Easier to integrate with other technologies
- Increased Business Agility

*But I've got this huge legacy code base -
how do I move to this connectivity model?*

Typical Client / Server application
w/ BL and UI mixed together



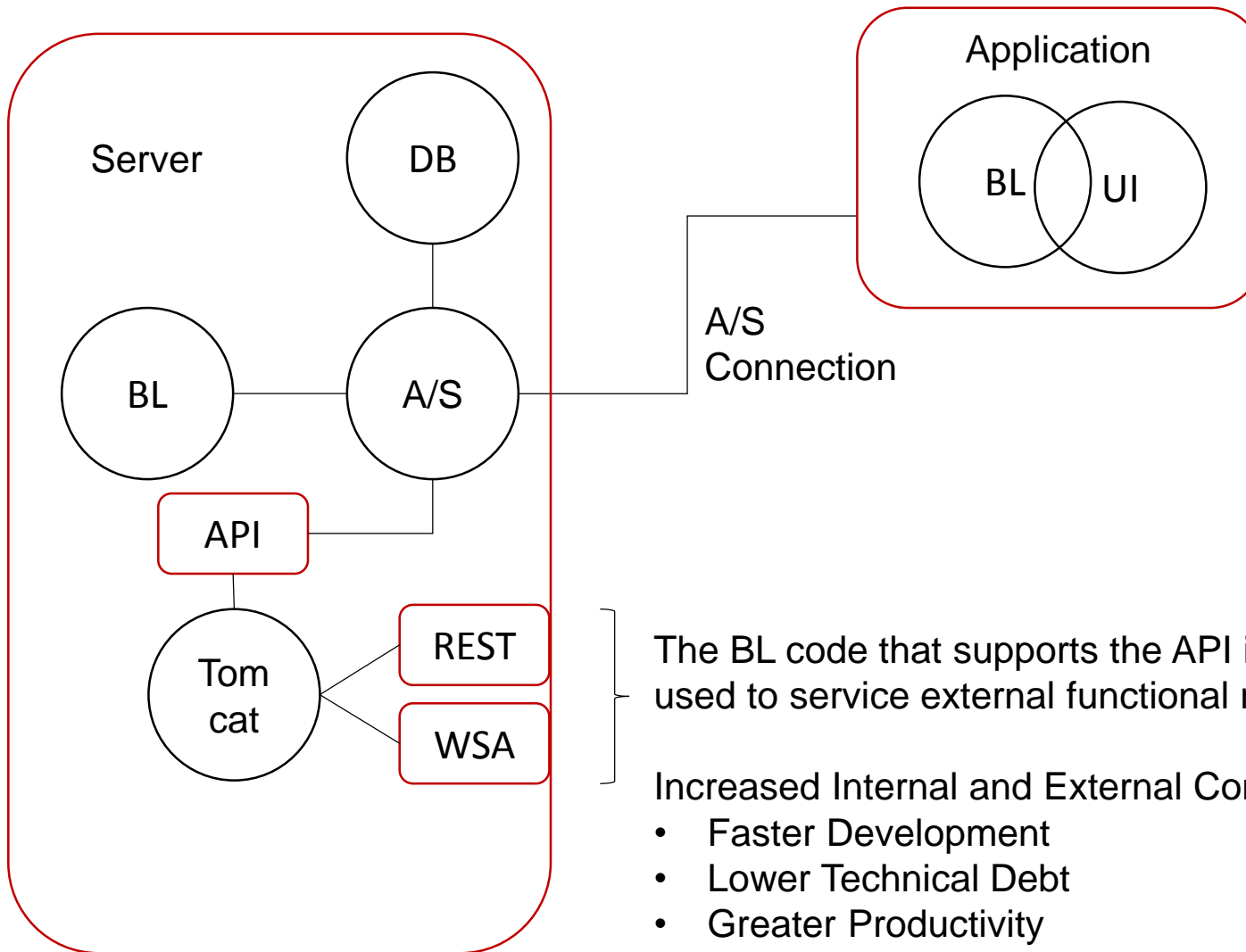
Goal - Migrate from C/S to an Appserver



Migrating from C/S to Appserver:

1. Create A/S Connection
2. Design a BL service for a given activity
3. Create BL logic behind A/S connection
4. Replace BL in Application with call to A/S logic
5. Replace all DB references w/ProDatasets
6. Call A/S logic to manage PDS data
7. Remove C/S db connection

From Client-Server to Appserver



The BL code that supports the API interfaces can also be used to service external functional requirements.

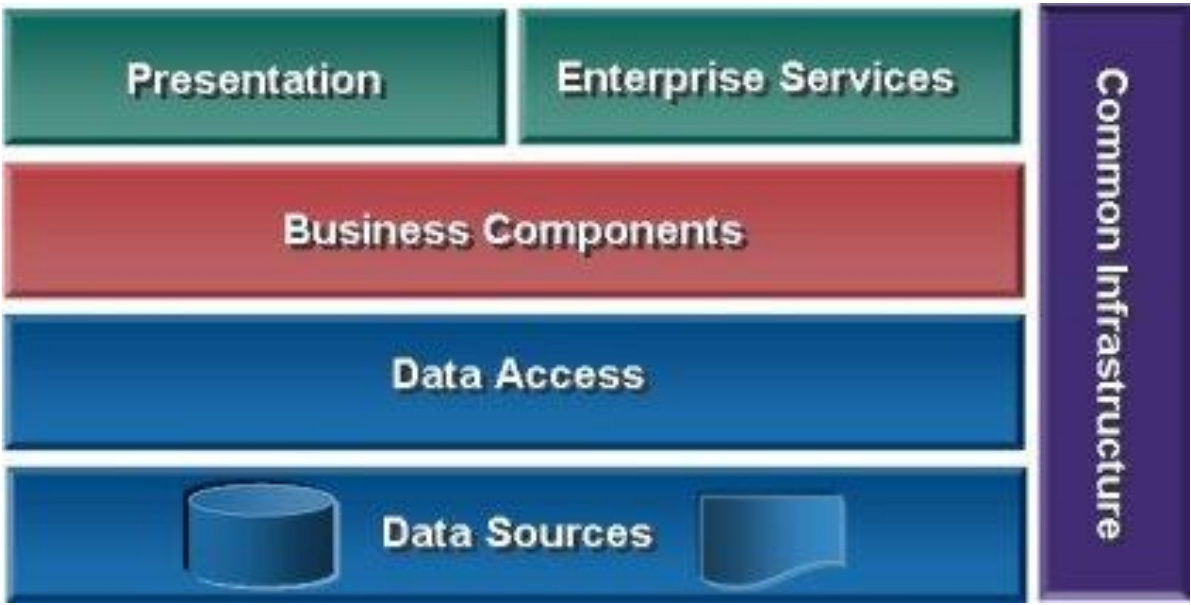
Increased Internal and External Connectivity Capabilities =

- Faster Development
- Lower Technical Debt
- Greater Productivity
- Increased Business Agility

What Does a Good API Look Like?

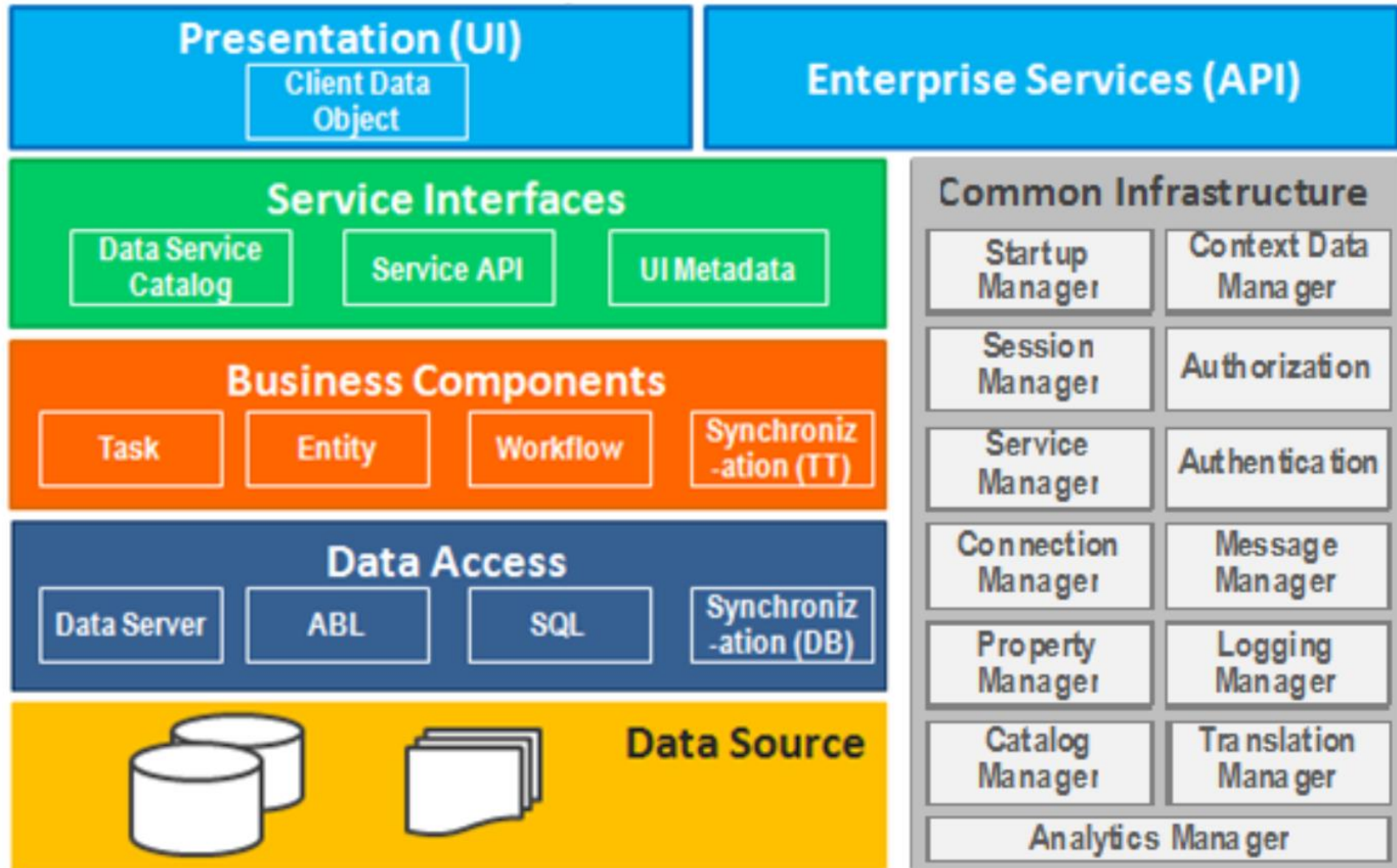
- Easy to learn
- Easy to use, even without documentation
- Hard to misuse
- Easy to read and maintain code that uses it
- Sufficiently powerful to satisfy requirements
- Easy to extend
- Appropriate to audience

Progress Open Edge Reference Model



Architectures – Progress OEAA

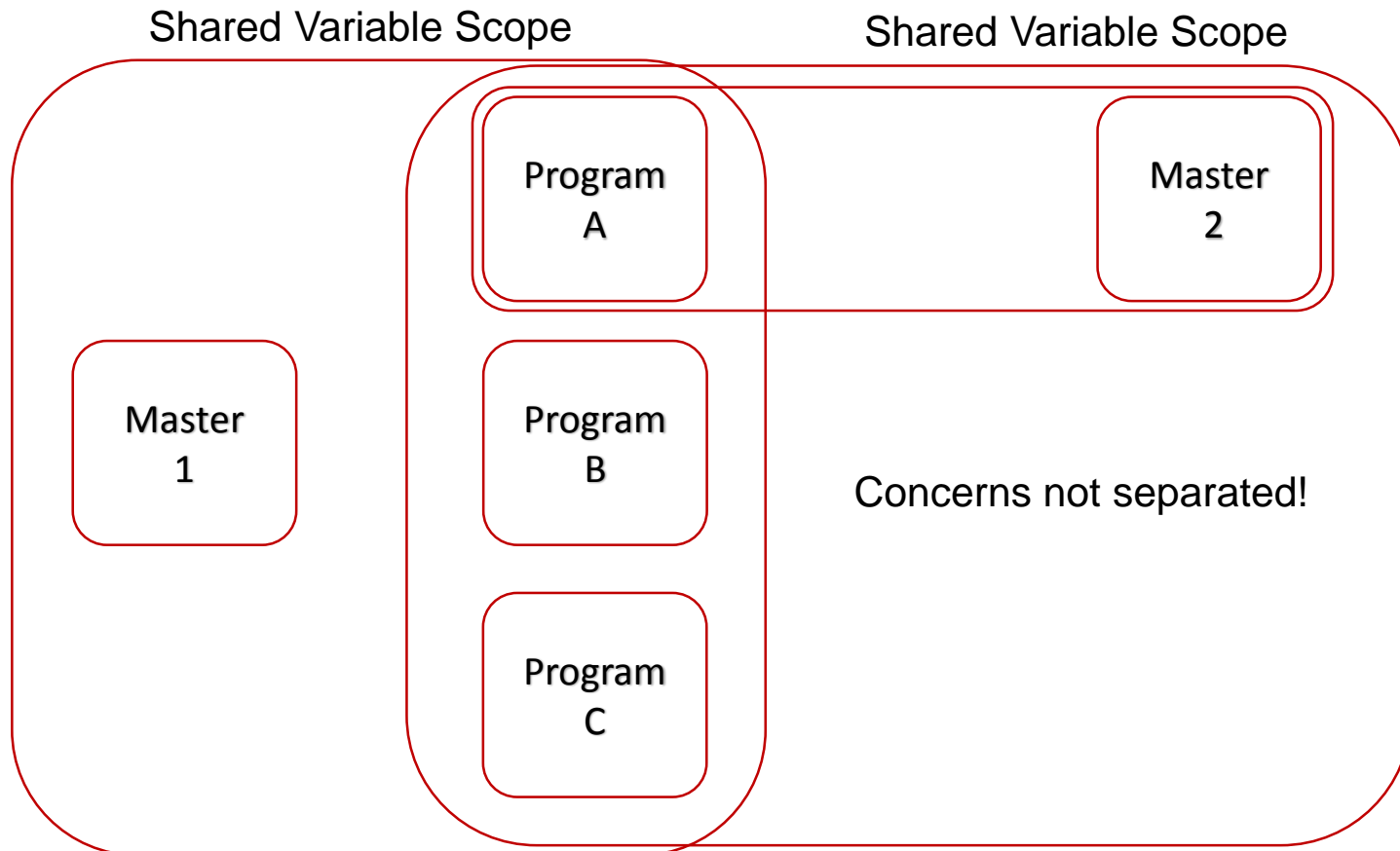
Common Component Specification Project Open Edge Application Architecture



Overcoming Shared Variables

The Problem – Overlapping Scope

- Master #1 uses BL in "A", "B", and "C"
- Shared variables used to transfer information
- Master #2 needs "A"s BL
- Can't just declare "A"s shared vars
- Needs to specify all Shared Vars – including ones not used



The Solution – Parameters and Classes

```
/* Shared Variables */  
DEFINE {1} SHARED VARIABLE ApplicationID AS INTEGER NO-UNDO.  
DEFINE {1} SHARED VARIABLE ActualUserID AS INTEGER NO-UNDO.  
DEFINE {1} SHARED VARIABLE EffectiveUserID AS INTEGER NO-UNDO.
```

```
/* SharedClass.i */  
DEFINE {1} VARIABLE oSharedClass AS SharedClass NO-UNDO
```

```
/* SharedClass.cls */  
CLASS SharedClass:  
  
DEFINE PUBLIC PROPERTY ApplicationID AS INTEGER NO-UNDO.  
DEFINE PUBLIC PROPERTY ActualUserID AS INTEGER NO-UNDO.  
DEFINE PUBLIC PROPERTY EffectiveUserID AS INTEGER NO-UNDO.  
  
END CLASS.
```

Overcoming Shared Variables

The Solution – Parameters and Classes

```
/* Shared Variable */  
{SharedVariables.i NEW}  
ApplicationID      = 1.  
ActualUserID      = 2.  
EffectiveUserID    = 3.
```

RUN ProgramA.p.

```
/* OO Based Shared Variable */  
{SharedClass.i NEW}  
  
oSharedClass = NEW SharedClass().  
  
oSharedClass:ApplicationID      = 1.  
oSharedClass:ActualUserID      = 2.  
oSharedClass:EffectiveUserID    = 3.
```

RUN AProgram.p.

- Concerns are better separated
- Overlapping scope reduced to one variable
- Encapsulation = add functionality to oSharedClass w/out unintended external interactions.
- Will require replacing all shared variable references with class references

The Solution – Parameters and Classes

```
/* OO Based Parameter */  
DEFINE VARIABLE oSharedClass AS SharedClass NO-UNDO.  
  
oSharedClass = NEW SharedClass().  
  
oSharedClass:ApplicationID      = 1.  
oSharedClass:ActualUserID      = 2.  
oSharedClass:EffectiveUserID    = 3.  
  
RUN ProgramA.p(oSharedClass).
```

- Concerns are better separated
- Issues with overlapping scope eliminated
- Encapsulation = add functionality to oSharedClass w/out unintended external interactions.
- Will require replacing all shared variable references with class references

```
{IncludeBL.i}
DEFINE VARIABLE iVarible AS INTEGER NO-UNDO.

FUNCTION CalcAValue
  RETURNS DECIMAL(iSourceNumber as INTEGER).
  /* Stuff */
RETURN(aValue).
END FUNCTION.

PROCEDURE LocalProcedure:
  /* Stuff */
END PROCEDURE.
```

- Variable definition scope bleeds over into enclosing code
- All too easy to have unintended variable interactions
- Requires recompiling all child programs when changed
- Does not separate concerns

```
CLASS IncludeBLCClass:

DEFINE VARIABLE iTempValue AS INTEGER NO-UNDO.

METHOD PUBLIC DECIMAL CalcAValue (iSourceNumber as INTEGER).
    /* Stuff */
RETURN(aValue).
END FUNCTION.

METHOD PUBLIC VOID LocalProcedure():
    /* Stuff */
END METHOD.

END CLASS.
```

- Variable definition scope contained in class
- Eliminates inadvertent interaction with enclosing code
- Does not require recompiling all child programs when internals are changed
- Separates concerns

```
/* Include BL Code */
{IncludeBL.i}
DEFINE VARIABLE iCalcValue AS INTEGER NO-UNDO.
ASSIGN iCalcValue = CalcAValue(1234).

/* Class BL Code */
DEFINE VARIABLE oIncludeBL AS IncludeBLClass NO-UNDO.
DEFINE VARIABLE iCalcValue AS INTEGER NO-UNDO.

oIncludeBL = NEW IncludeBLClass().

ASSIGN iCalcValue = oIncludeBL:CalcAValue(1234).

oIncludeBL:LocalProcedure().
```

Analytics uses

- mathematics,
- statistics,
- descriptive techniques,
- machine learning, and
- predictive models

to gain knowledge from data. These insights can be used to recommend action or to guide decision making rooted in business context.

Uses of Analytics:

- Customer Service Personalization
- Product Recommendations
- Warehouse Product Placement Optimization
- Warehouse Inventory Management Optimization
- Retail Stock Purchasing
- Marketing Response Measurement
- NeoNatal Healthcare
- Financial Fraud
- Law Enforcement
- Customer Churn prediction

And there's more! <https://www.informs.org/Sites/Getting-Started-With-Analytics/Analytics-Success-Stories>

Georgia Institute of Technology regularly hosts a series of on-line courses.

Students attending these courses generate thousands of routine messages for each course and is a heavy load for the TAs that need to answer their queries.

Recently Georgia Tech hired Jill Watson to help answer student questions. Jill did such a good job that a student at one of Jill's courses wanted to nominate Jill as an outstanding TA.

There's only one thing...

Jill is powered by IBM's Watson analytics system.

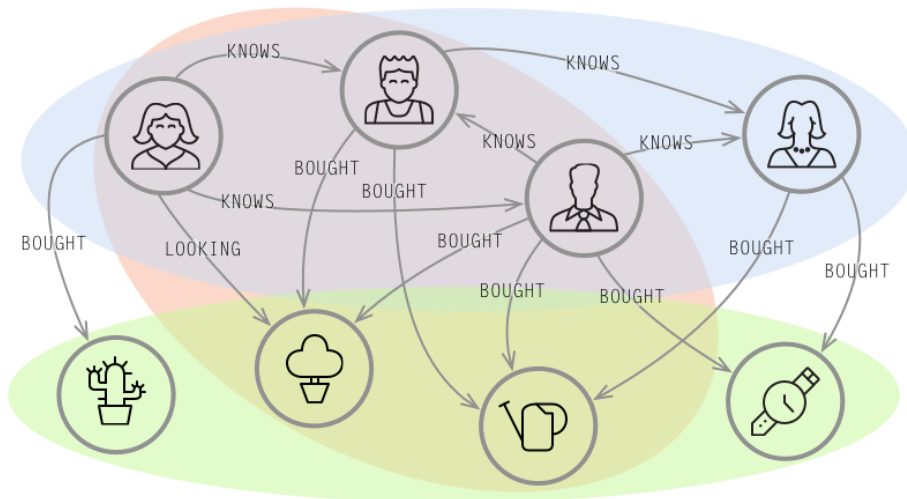
Though machine learning has long been part of Google's technology ... the company circa 2016 is *obsessed* with it.

CEO Sundar Pichai laid out the corporate mindset:

“Machine learning is a core, transformative way by which we’re rethinking how we’re doing everything. We are thoughtfully applying it across all our products, be it search, ads, YouTube, or Play. And we’re in early days, but you will see us — in a systematic way — apply machine learning in all these areas.”

Netflix says Geography, Age, and Gender are “Garbage” for Predicting Taste

...if you want to ... get someone to stream more of your content, you're better off leveraging what you know about similar individuals in completely different demographic groups, than trying to cater to broad generalizations...



Progress OpenEdge
+
REST
+
Graph Database
=
Recommendation Engine
Fraud Detection
Security Access Control

In the era of big data, consumer profiling ...[needs to] get past the surface and see what really makes them tick.

<http://fortune.com/2016/03/27/netflix-predicts-taste>

Because Nobody is Safe

A Disruptive Company –

- Changed an entire industry on its way to becoming a global brand and a verb
- HQ is centralized in a single city away from most vendors
- Has centralized control of vendor rates
- Set prices for services below rate where vendors can make a reasonable living
- Vendors are not happy.....

Signs you're about to be disrupted:

1. Venture capitalists are circling
2. You're using technology to cut costs, rather than create value
3. Your customers are getting older
4. Your service is very high-touch
5. Your customer satisfaction is low

Eating the Disruptors' Eggs

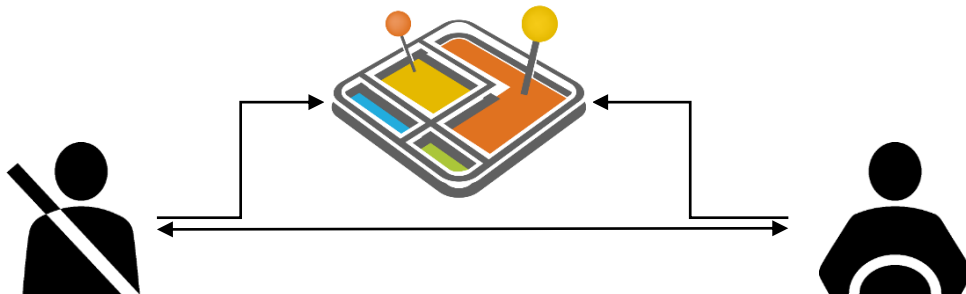
Nobody is Safe



Uber Model

- Passenger and Rider subject to Uber's rules
- Centralized Administration
- Centralized Control
- "Big Business" model

Arcade City



Next Gen Ridesharing

- Uses Arcade City to connect
- Completes TX Peer to Peer
- Uses blockchain / ethereum technology for contracts and trust
- Makes the middleman optional

Eating the Disruptors' Eggs

Black Market Ride-Sharing Explodes In Austin After Voters Drive Out Uber And Lyft

Now that Uber and Lyft have pulled out of Austin due to onerous new city regulations, drivers and riders are turning to black market ride-sharing.

<http://thefederalist.com/2016/05/23/black-market-ride-sharing-uber-lyft/>

There are countless thousands of angry Uber and Lyft drivers looking for an alternative. We'll start by building what they want, then move in the direction that we know will maximally benefit both drivers and customers over the long term: *decentralization on the blockchain.* ... **What SMTP is to email, Arcade City will become for distributed logistics.**

<http://cointelegraph.com/news/arcade-city-decentralized-blockchain-based-answer-to-uber>



- Direct Peer to Peer Business
- Decentralized Administration
- Decentralized Control

Here's the big idea ... cut out the corporate middlemen ... *by transparently providing rider and driver with clear information about the other party to each transaction, including a strong reputation and ratings system where riders and drivers 'level up' after community-vetted good behavior on the platform.* <http://arcade.city/>

Questions?



Related Sessions

252: Case Study: Application Modernization
Rob Marshall, Progress Software

484: Give Your Project a Spark
Dustin Grau, Progress BravePoint

798: Common Component Specification A deep dive into the specs
Mike Fechner, Consultingwerk

871: Facing the Challenges of modernizing legacy OpenEdge code and concurrency control
Oscar Perez Cedron and Sophy Nathalian, Apero Solutions

961: A Modernization Platform for the Web
Shelley Chase, Progress Software

258: Hybrid Mobile: It's Not about Cars!
Ricardo Perdigao & Jean Munro, Progress Software

799: REST In Piece - Mastering the JSDO with a dynamic ABL backend
Mike Fechner, Consultingwerk

873: Standardizing OpenEdge Application Frameworks through Community Authored Specifications.
Tom Kincaid, Mike Fechner, Shelley Chase, Christopher Longo Peter Judge,
Paul Moberg, Mike Jacobs