



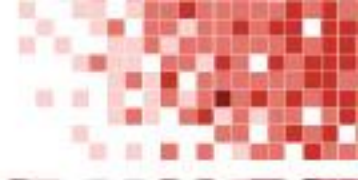
Adding order and cohesion to disjointed online services with Open Source Portals

Alistair Oldfield – Emeldi Group - Official Liferay Partner



April, 2013

Agenda



- Introduction to a common problem
 - Examples of concept in real life
 - Challenges facing most portal landscapes today
- Assessing the goals and defining an approach
 - Ideals
 - Goals
- Approach
 - Consolidating Presentation Layer - Liferay
 - Consolidating Underlying Systems
 - Identity Management
 - CMS
 - ESB
- Conclusion
- Q&A

Toronto, Canada

Est. 1834, Pop. 5.5 M

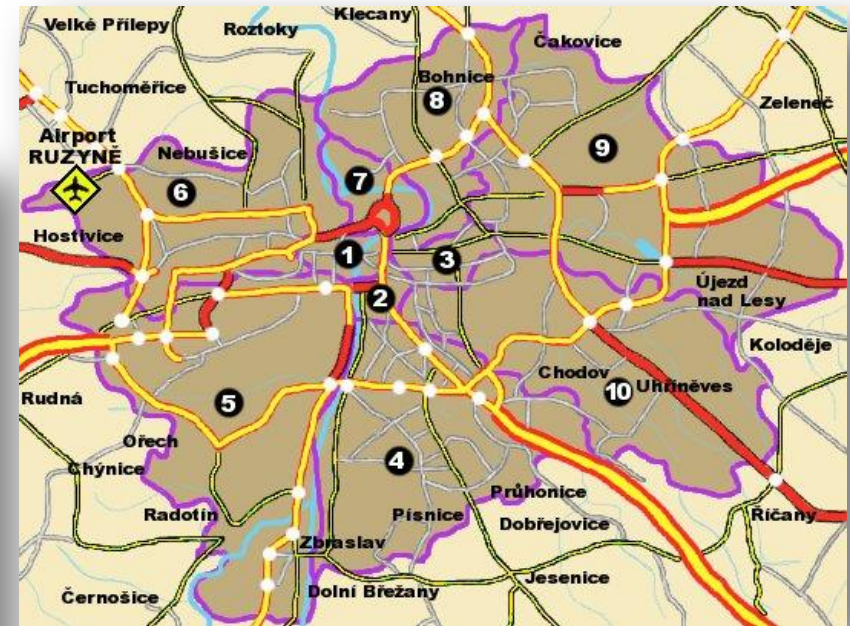
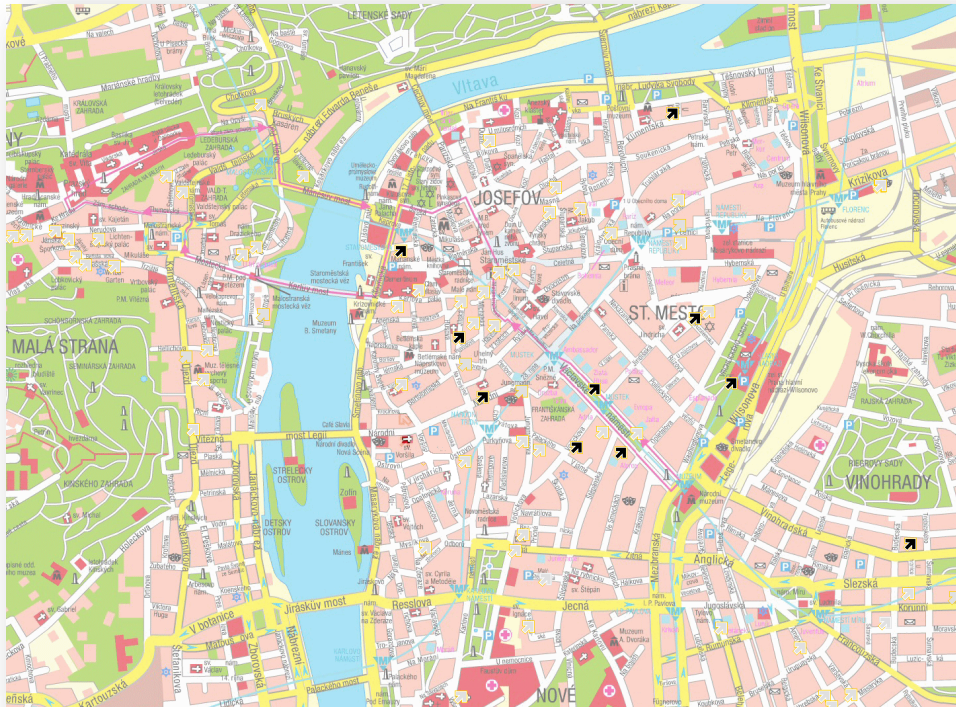
- Example of relatively modern city planning techniques
- Grid pattern (N-S, E-W)



Prague, Czech Republic

Est. 885, Pop. 1.1 M

- Example of legacy city planning
- Ad-hoc network



1000 years of advances

- Building architecture & engineering

- Most structures built with minimal consideration beyond short-term future and immediate survival

vs.

- Houses passed from one generation to another, built on established building practices



- Forms of government & law

- Decentralization of municipal concerns



- Logistics/transportation techniques

- Advances in transport required advances in city infrastructure and roads

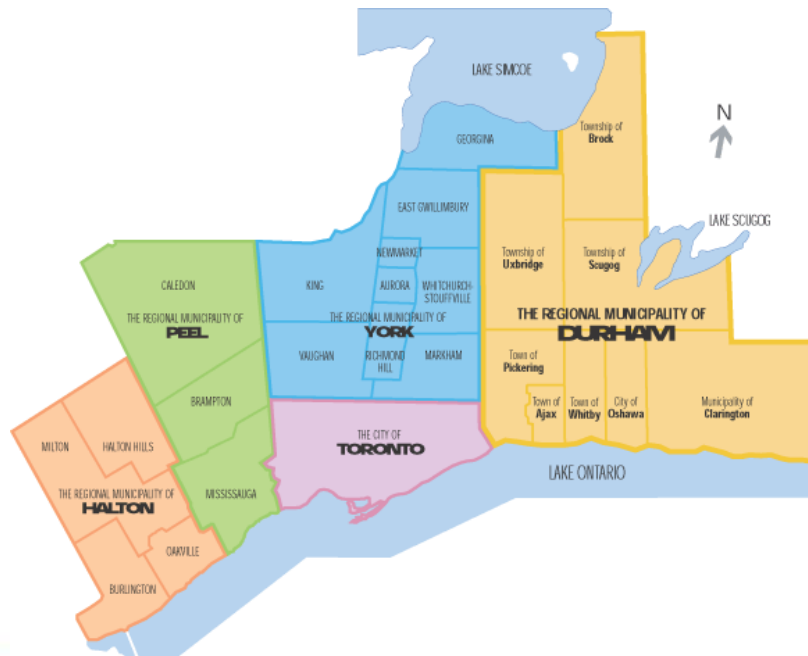


- Advances in commerce

**All significantly impacting
urban planning and
governance**

Result in terms of Municipal Architecture/Topology

- Highly centralized municipal governance
 - Serves 5.5 M people
 - Clean Grid delineation
- Ad-hoc, stop-gap municipal governance
 - Serves 1.1 M people
 - Non-centralized
 - Disjoint services

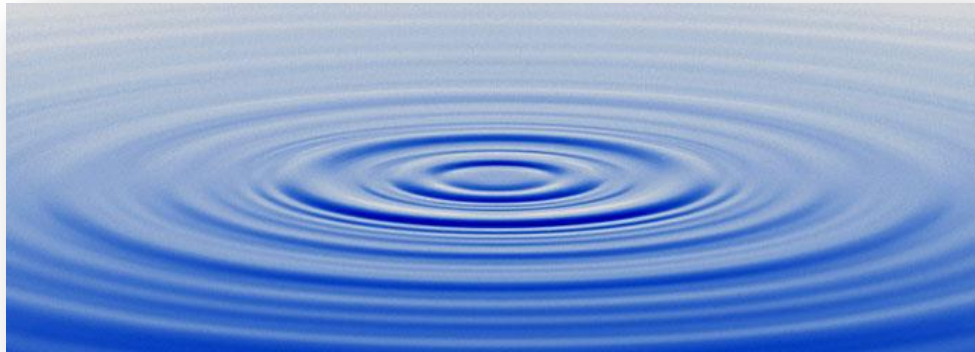


VS.



Cause & Effect

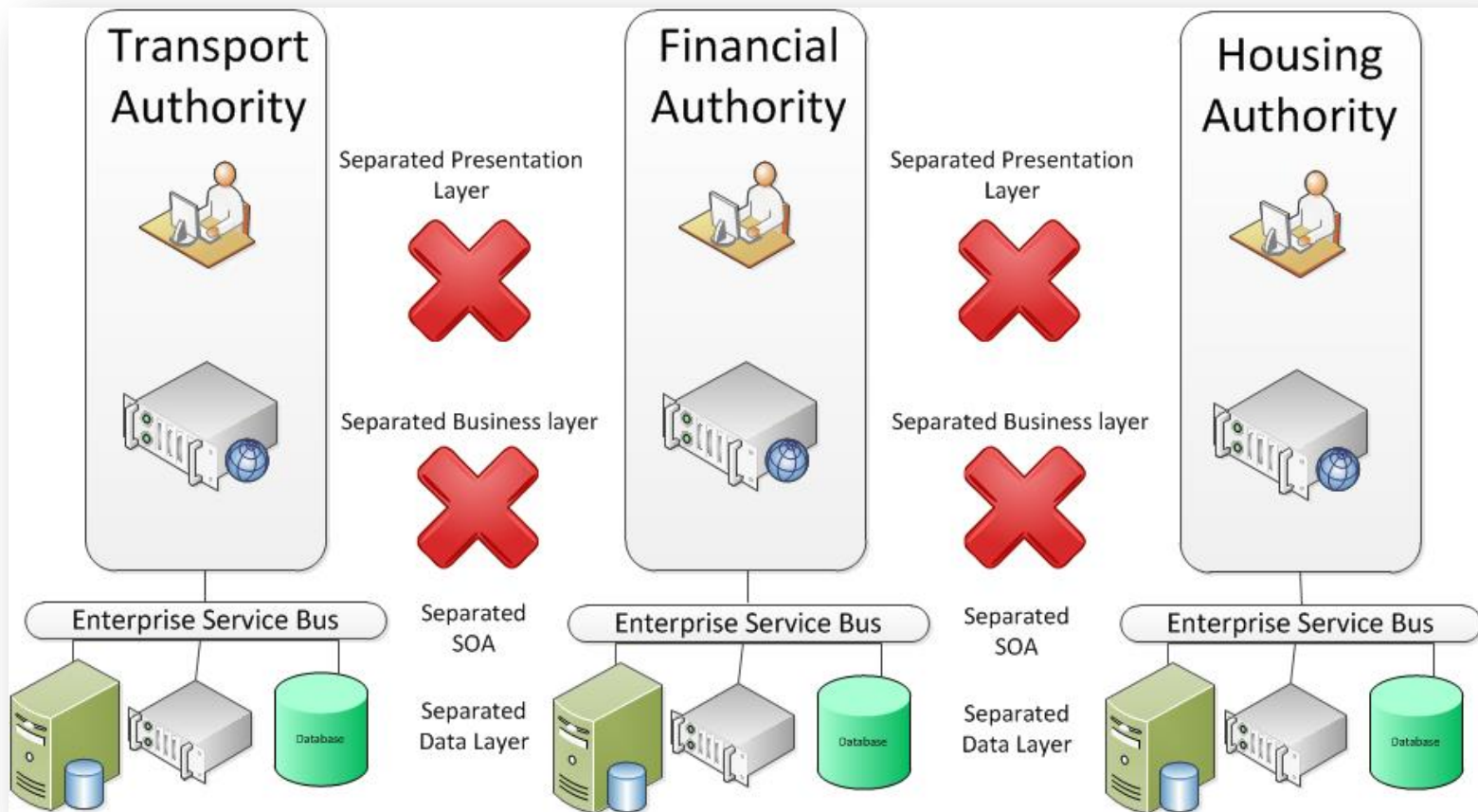
- City is build on an entirely different set of needs
 - Prague tailored to satisfy a set of needs spanning >1000 years
 - Toronto tailored to satisfy a set of needs spanning <200 years



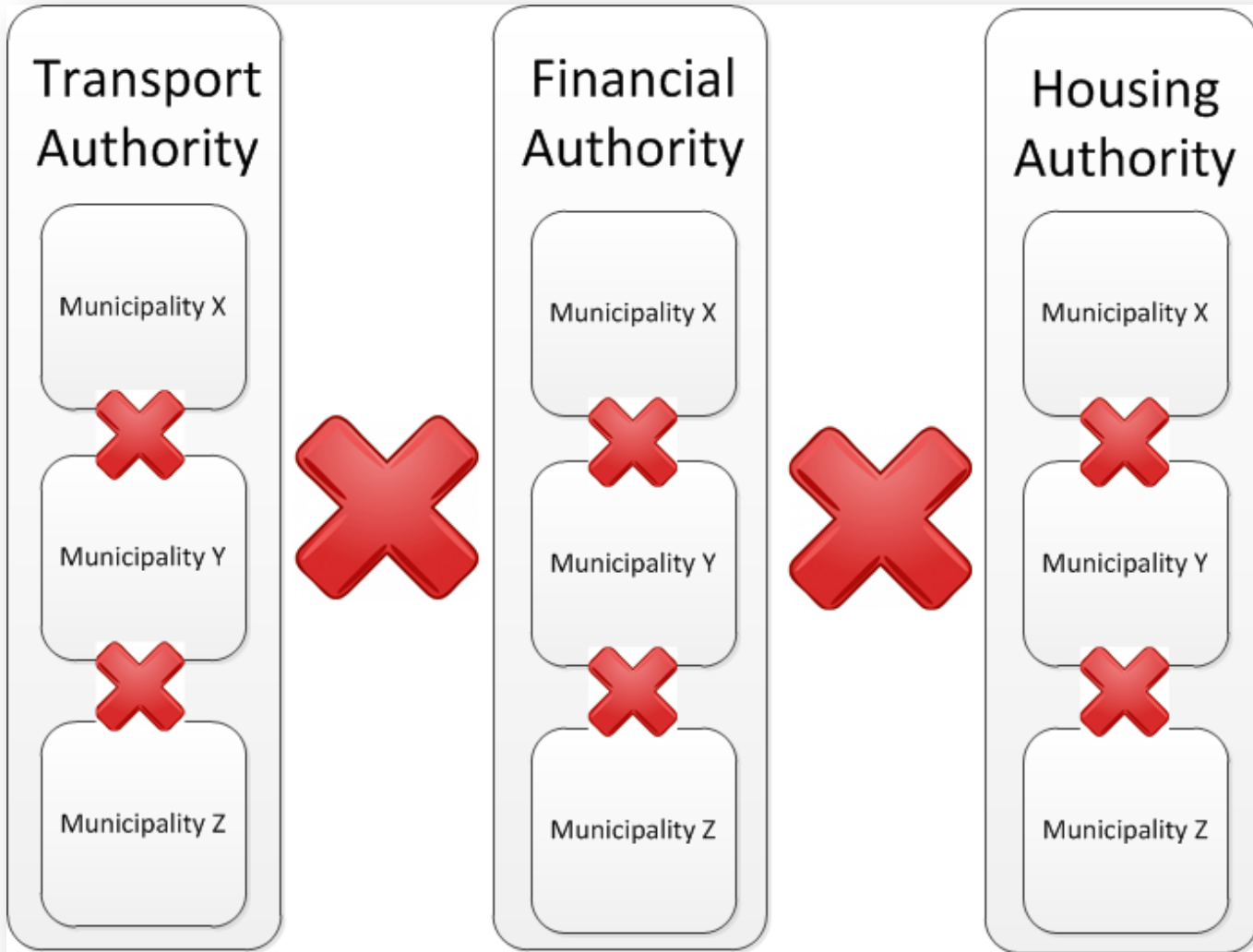
- As both cities evolved throughout time in order to adapt to 2013, Toronto has had far less distance to travel than Prague
- AS-IS city topologies vary significantly as a result
- Obvious conclusion: Newer cities are more adapted to today's requirements thanks older cities

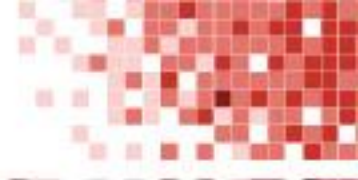
Disjoint Architectures

- City topology & online government services have evolved in a very similar way and face very similar challenges
- Czech Government Portal Landscape:

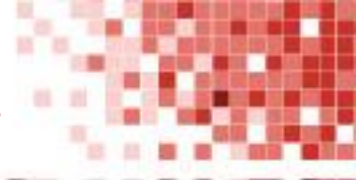


Internal Disconnect





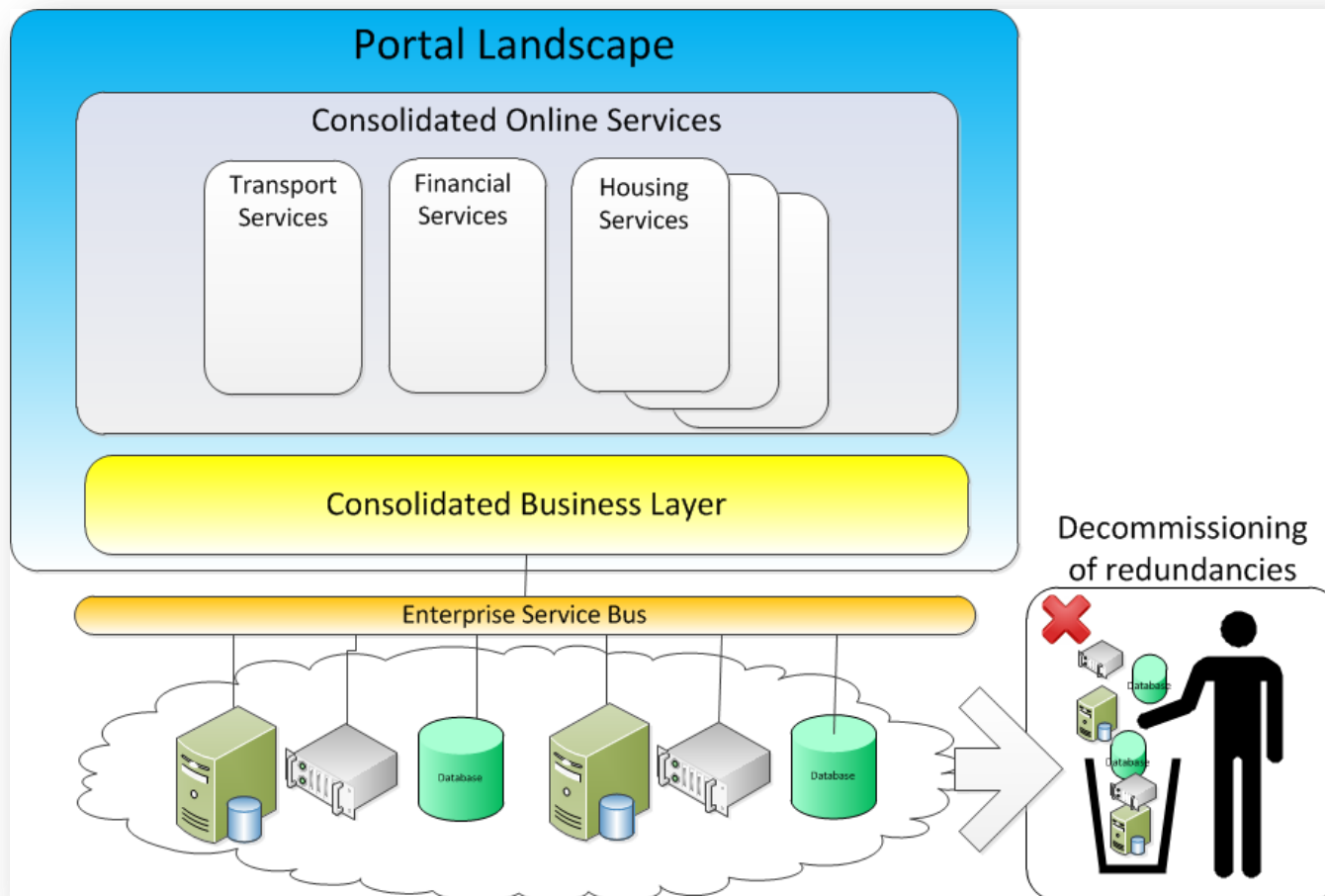
- Change of address
 - Worst possible thing to do
- Providing proof of X based on data which is readily available in one authority, however not to another
 - Stamped hardcopy still required in many cases
- Trade licensing
 - Registration in more than 1 authority providing data which should be readily available
- Citizens can only visit relevant authorities located in the municipality of their permanent address



- Information Systems
 - Duplication of TCO across all IT systems
 - Duplication and increased work load on specialized staff
 - Duplication of HW, SW licenses
 - Difficult to introduce new automated business processes and online services
 - Data duplication, redundancies, inconsistencies
- Business Processes
 - Manual / slow (paper work)
 - Redundant
 - Chaotic
- User point of view
 - Must interact with many sites / interfaces
 - Difficult to find desired services

Utopian end-to-end Architecture

- How to get out of this situation?
- Complete overhaul and consolidation of portal landscape?
 - Let's face it, not likely...



Change is never easy



ICT point of view

- Complete overhaul and consolidation of IT systems
- “Best” in terms of architectural organization & cleanliness
- “Best” in terms of OPEX (potentially at least)



Business point of view

- But is it the “best” in terms of realized business benefits vs. cost?
- Time constraints
- Cost of overhaul would be astronomical (high APEX)
- Is there a cheaper way?



Customer point of view

- Difficulty using online services translates to me not using them at all
- “I don’t care how you do it, just make it work better”

How to get ourselves out of this situation?



Goals

In order to find an optimal solution, let's identify goals:

1

Automate
as many services
as possible

- **Reduces cost** across the board
 - call centres
 - municipal offices
- Productivity loss for tax payer: time spent in municipal office/phone



2

Simplify access
to online services

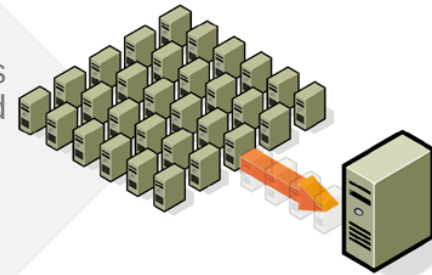
- If the user cannot find the services, he/she will not use it.
- This means **consolidating endpoints** into a single portal (entry-point) providing access to all services
- Includes **SSO**

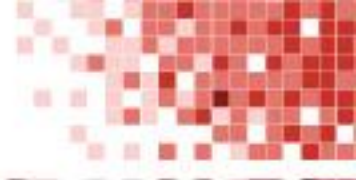


3

Consolidate
disjoint
information
systems

- Disjoint systems running similar or functionally-related business processes cost **money** in terms of both **APEX** and **OPEX**
- Improve **QoS** by reducing inconsistencies/redundancies in business processes & data





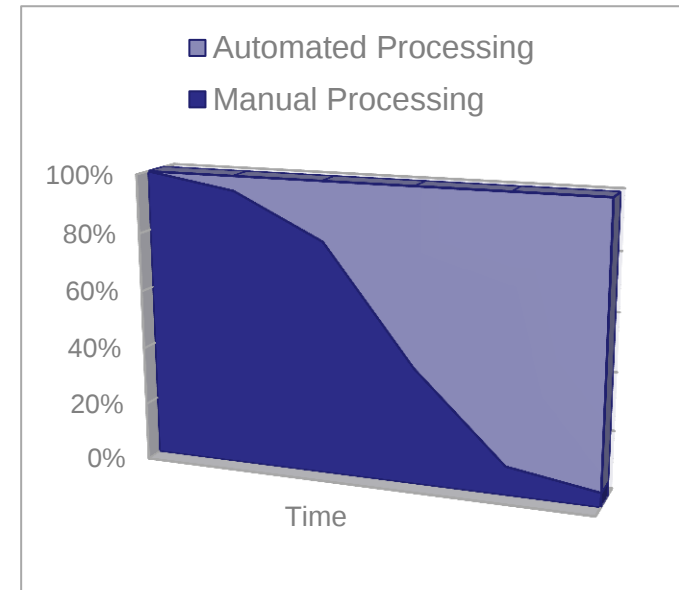
4

Execute the
change in a
cost-effective
manner

- Cost will be **key deciding factor** in defining approach
- Project will not initiate without realizing this goal **above all else**

Goal #1: Automate

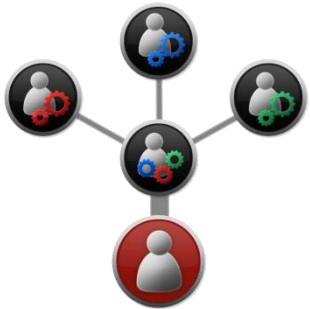
- Create a platform into which new services can be easily plugged-in as portal landscape evolves
- Platform should focus on supporting many services which do 1 thing, but do it well (mini-apps)
- Choose a technology set and standards to be used across the board to allow for smooth interoperability between underlying applications
- Platform to allow for gradual introduction of additional automated services over time
 - Initially can be rolled-out quickly and then gradually enhanced with more functionality over time
 - Allows for adoption of agile enhancements vs. big-bang release



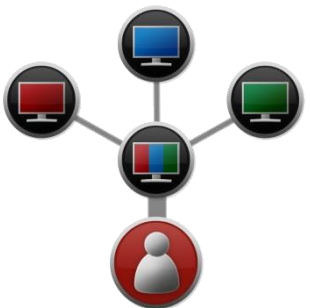
Goal #2: Simply Access



- Single Address / Single Entry
 - Introduce a single portal which will “contain” these services and serve them to customers from a single entry point
 - Reinforces the perception of a unified, fluid and personalized online experience for the customer
- Portal should treat a single individual as the same user across all services

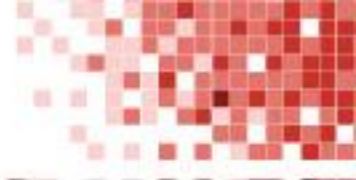


- Unifies customer profiles across the integrated portal environment
 - Include Single Sign On for any additional external portal (but we still want to reduce these)
 - Customer identity, preferences and settings are maintained, managed and shared across all integrated portals
- Unify the look & feel of the entire portal landscape



- Provide a consistent visual language and intuitive navigation to desired services
 - Eliminate the verticals in the presentation layer
 - The customer should not feel that they are interacting with discrete portals within the same organization

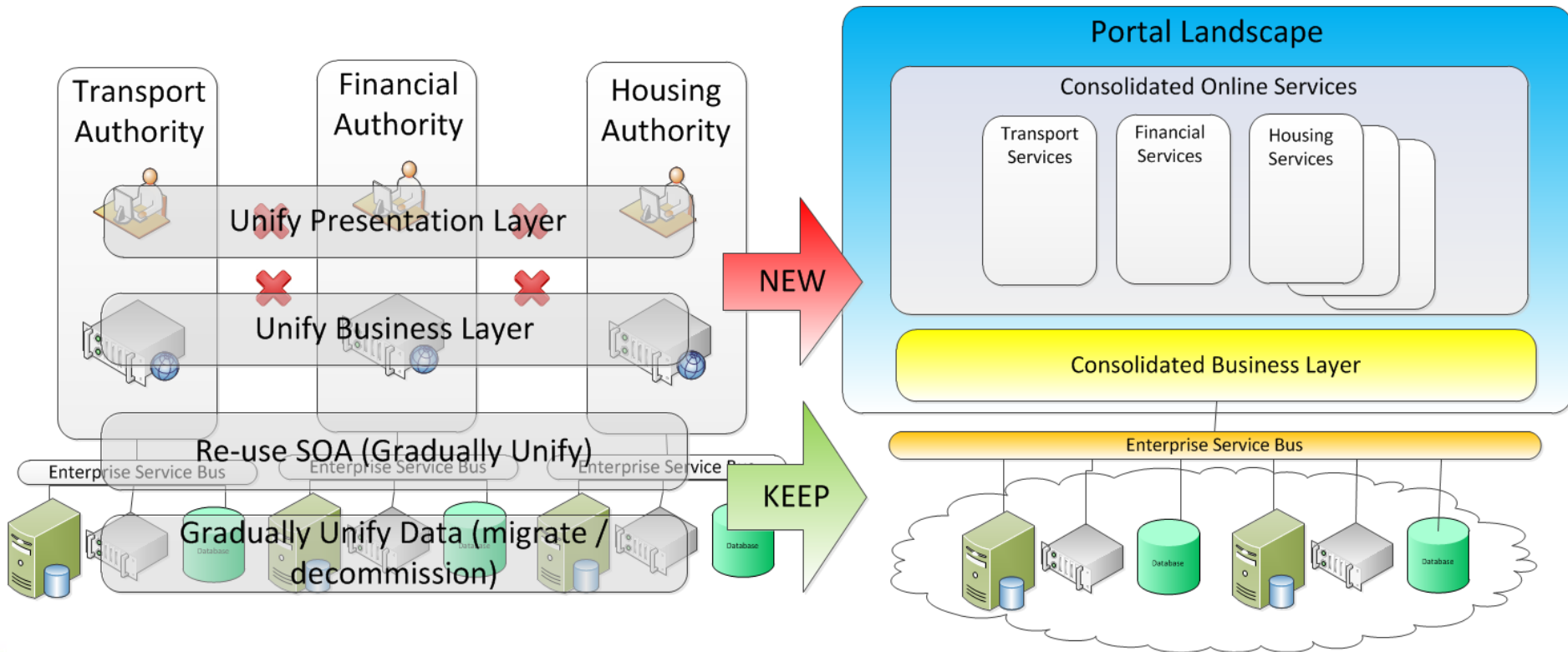
Goal #3: Consolidate



- Reducing the number of underlying systems is not a trivial task
- Identity management should be the initial focus
 - De-dup user base
 - Define a strategy to aggregate (virtually or physically) individuals' data
- Remaining information systems in SOA:
 - After user data aggregation we can focus on gradual phasing-out of redundant systems and data stores (or not)
 - 1) CMS: only necessary if legacy web content should be retained
 - This can be handled via migrations (more on this later)
 - 2) Business Processes: worth taking a look into identifying any duplications or distribution of functionally-related processes across multiple disjoint systems
- SOA & ESB: more later

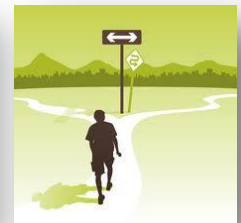
Approach – Phase 1

- Often preceded by a PoC
- Overhaul presentation layer ONLY – **Limit effort to this piece alone**
- Keep underlying SOA & Data
- Releasable for public consumption



Overhaul presentation layer

- Current situation: Multiple portal verticals
 - Different domains/sub-domains
- Many services may already exist online but are difficult for visitor to find!
 - IT investment made, benefits not being fully realised
 - Ultimately drives traffic away from online space and back into call centres
- Target: adopt a portal technology which:
 - Allows for combining of various disjoint web applications to co-exist within a set of pages and application container
 - Is affordable: (read **Open Source**)
- Consider:



What is Liferay Portal?

- Liferay portal is a Web-based application that aggregates services from different sources, and hosts the presentation layer of information systems
 - Search, personalization, single sign-on, content aggregation, etc..
- Portal pages may have different sets of portlets creating different content for users and providing a **consolidated** view for the visitor to these services.

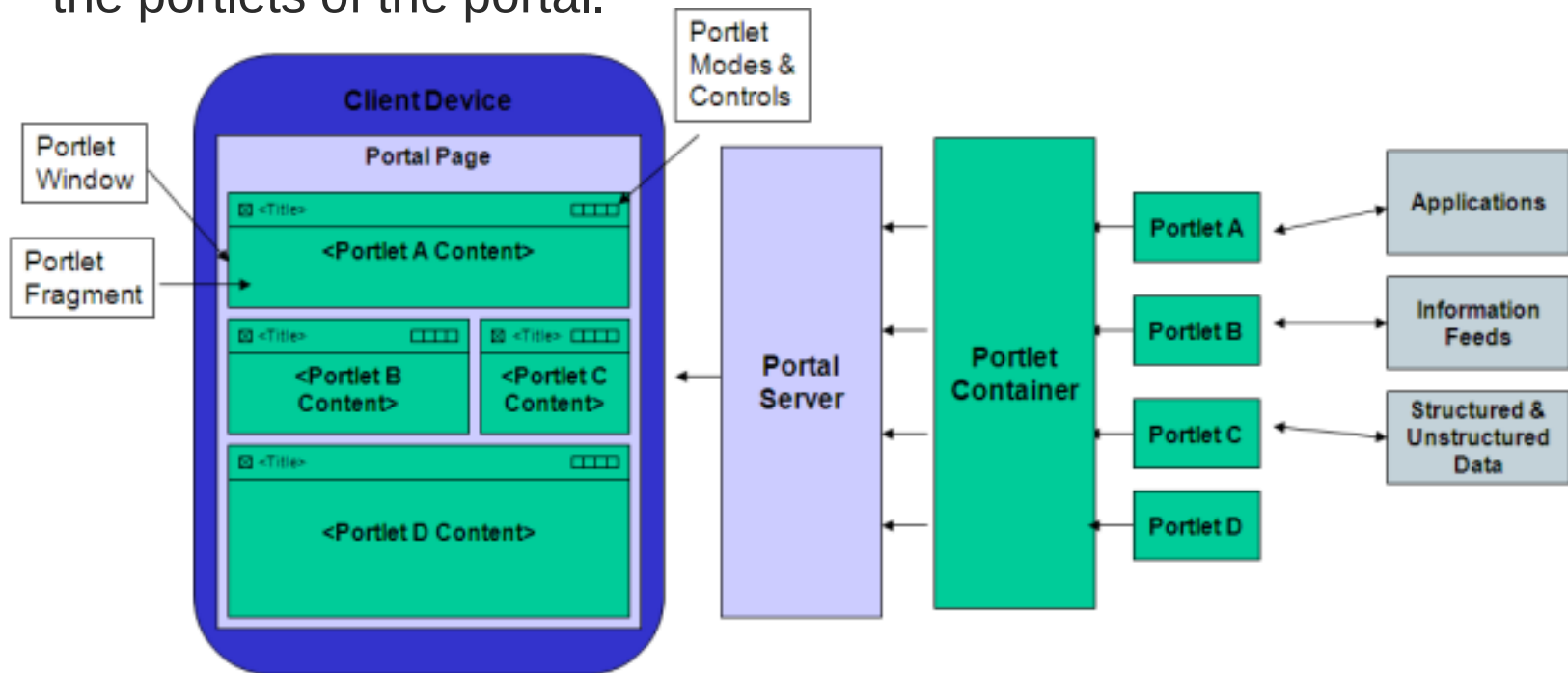


The screenshot shows the O2 Liferay Portal interface. The top navigation bar includes links for 'Pre Vás' and 'Pre Firmu'. Below this is a secondary navigation bar with links for 'Volania', 'Internet', 'Telefóny', 'Pomoc a podpora', 'Moje O2', and 'O2 Super zóna'. The main content area is divided into several portlets:

- Navigation portlet:** Located on the left side, it contains a search bar and a list of services.
- Search portlet:** Located on the right side, it contains a search bar and a list of services.
- Main Content Portlet:** The central area of the page, featuring a large banner for 'Sme fér vždy a ku všetkým' (We are fair always and to everyone) with a group of puppets. Below the banner are several promotional tiles for O2 services, including 'Telefóny s bonusom', 'Sme fér ku všetkým', 'Zľava 30% z faktúry', 'Internet do mobilu na 15 dní zadarmo', and 'Bonus na telefón s O2 Paušálom'.
- Chat portlet:** Located on the bottom left, it features a chat icon and a 'Začať chat' button.
- Marketing Portlets:** Located on the bottom right, they include promotional tiles for O2 services, such as 'Karta O2 Fér za 0 €', 'Zľava 30% z faktúry na 1 rok', 'Internet do mobilu na 15 dní zadarmo', and 'Bonus na telefón s O2 Paušálom'.

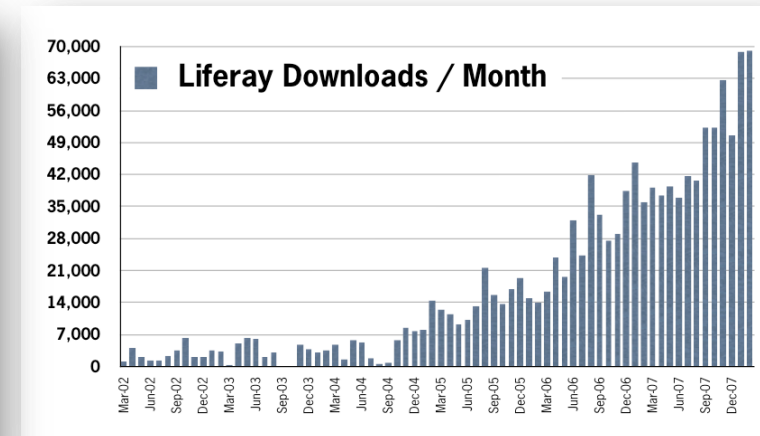
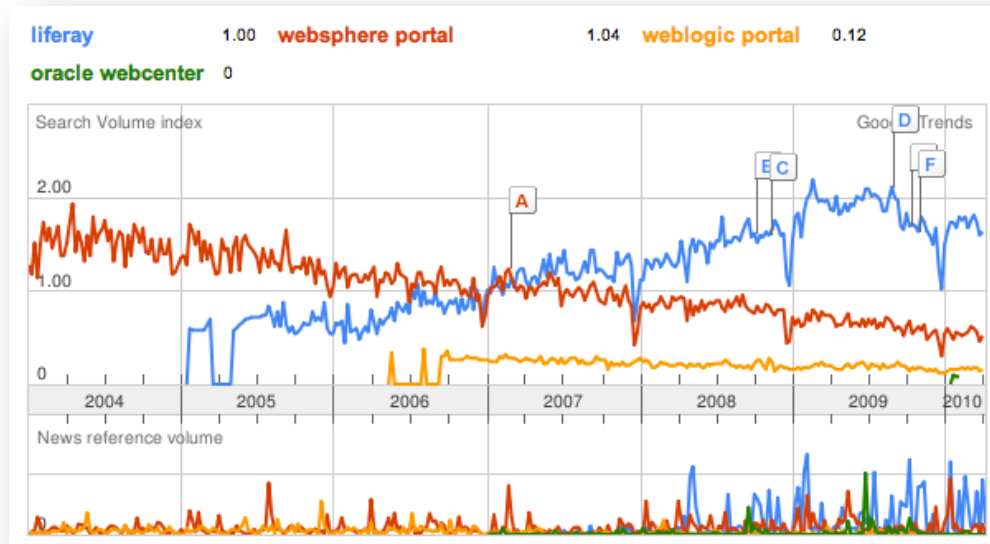
What is a Portlet?

- A portlet is a piece of Java code that manages the content of one section of a web portal's HTML.
- It can do anything else that a Java web application can do.
 - You can connect a portlet to a database, invoke a web service, download an RSS feed, etc.
- It lives in a **portlet container**, which creates, manages, and destroys all the portlets of the portal.



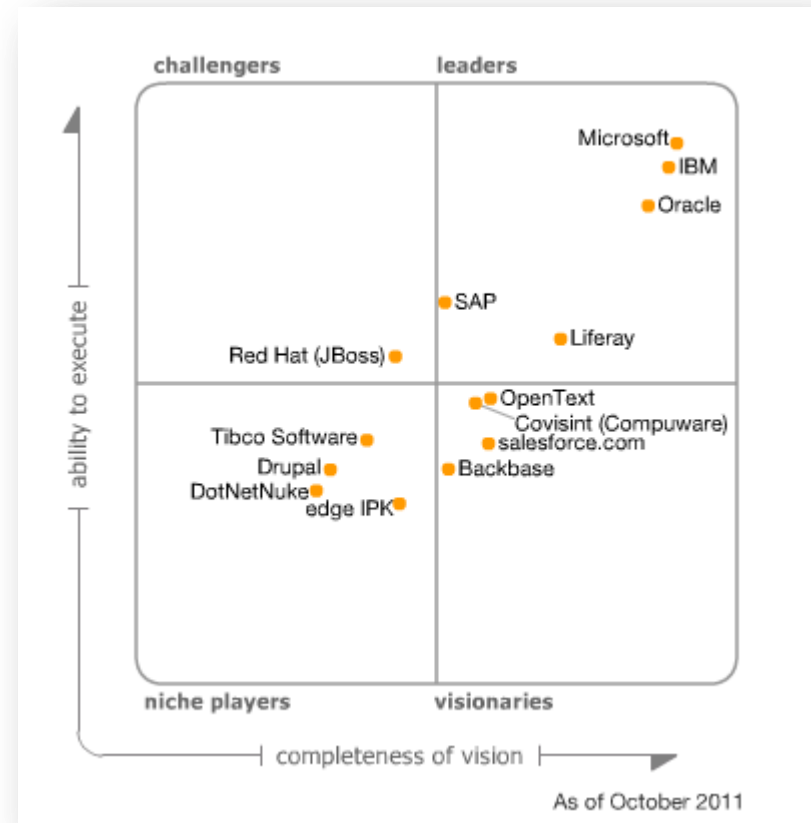
Liferay vs. The World

- Open alternative to:
 - IBM Websphere Portal
 - Oracle/BEA Portal
 - Sharepoint
 - Jive Clearspace for collaboration
- Liferay Portal is the most widely downloaded, Open Source Portal in the world
 - Over 1 Million downloads & over 60,000 downloads per month
 - Over 5800 registered community participants
 - Over 50 active community contributors



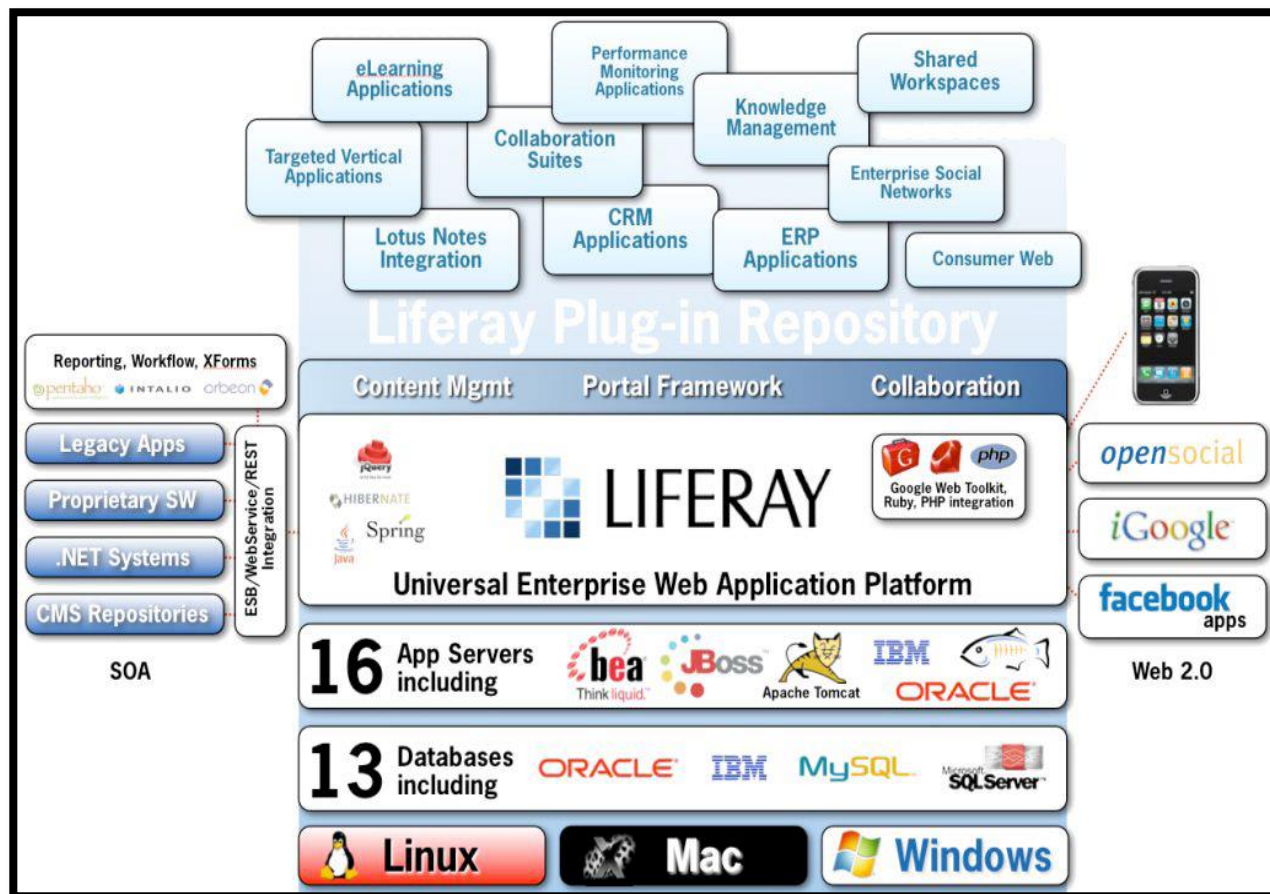


- Liferay has been a Leader in Gartner's Magic Quadrant for Horizontal Portal Products since 2010
- According to Gartner, Liferay is fastest growing player in portal technology



Liferay Strengths

- Open Source & with a thriving community built around it
 - Constant enhancements/improvements
 - Growing online resources and discussions
- Interoperability
 - All major OSs
 - All major DBs
 - All major ASs
 - JSR-168
- Enterprise Edition License
- Features!

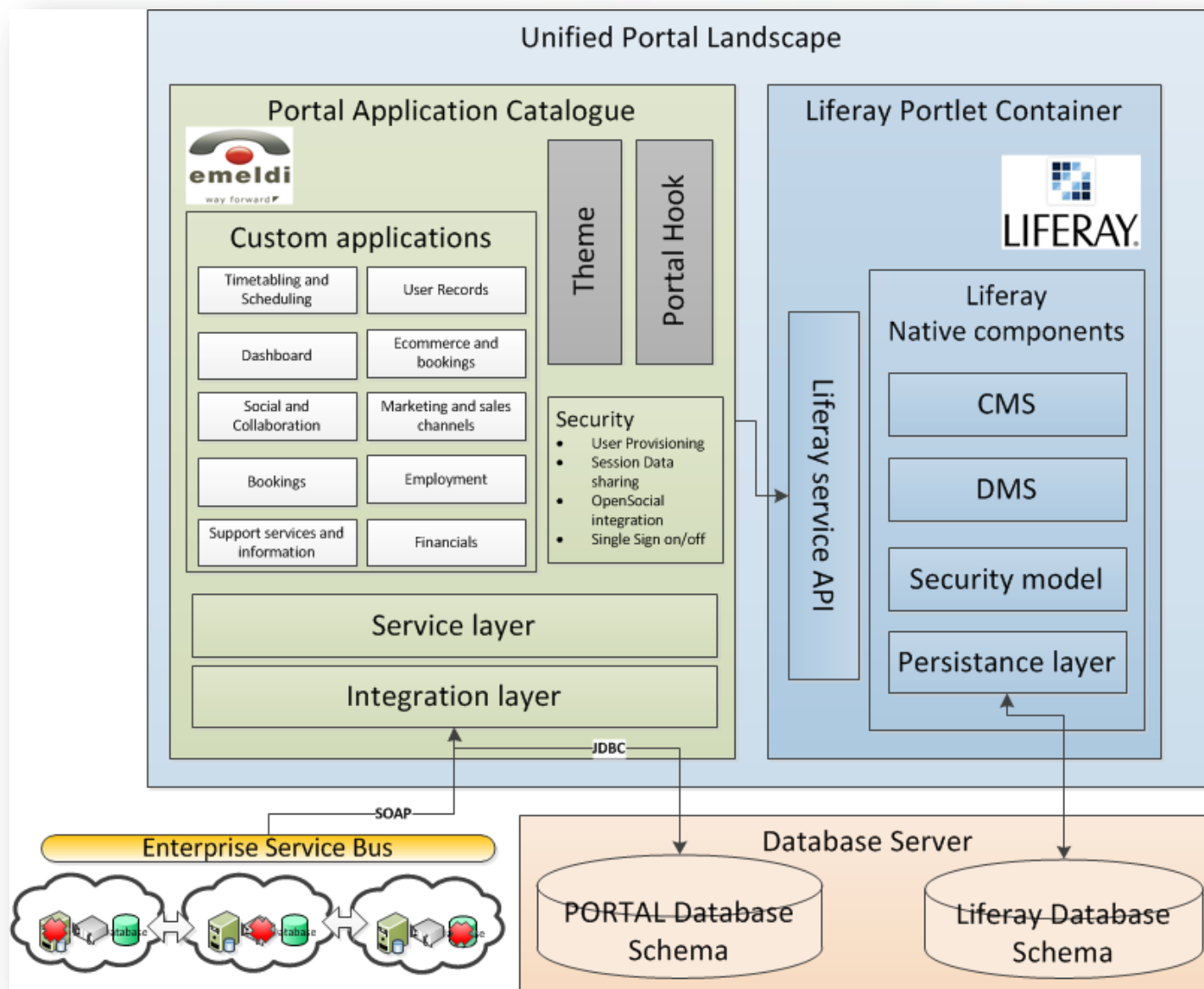


Liferay Features & Interoperability

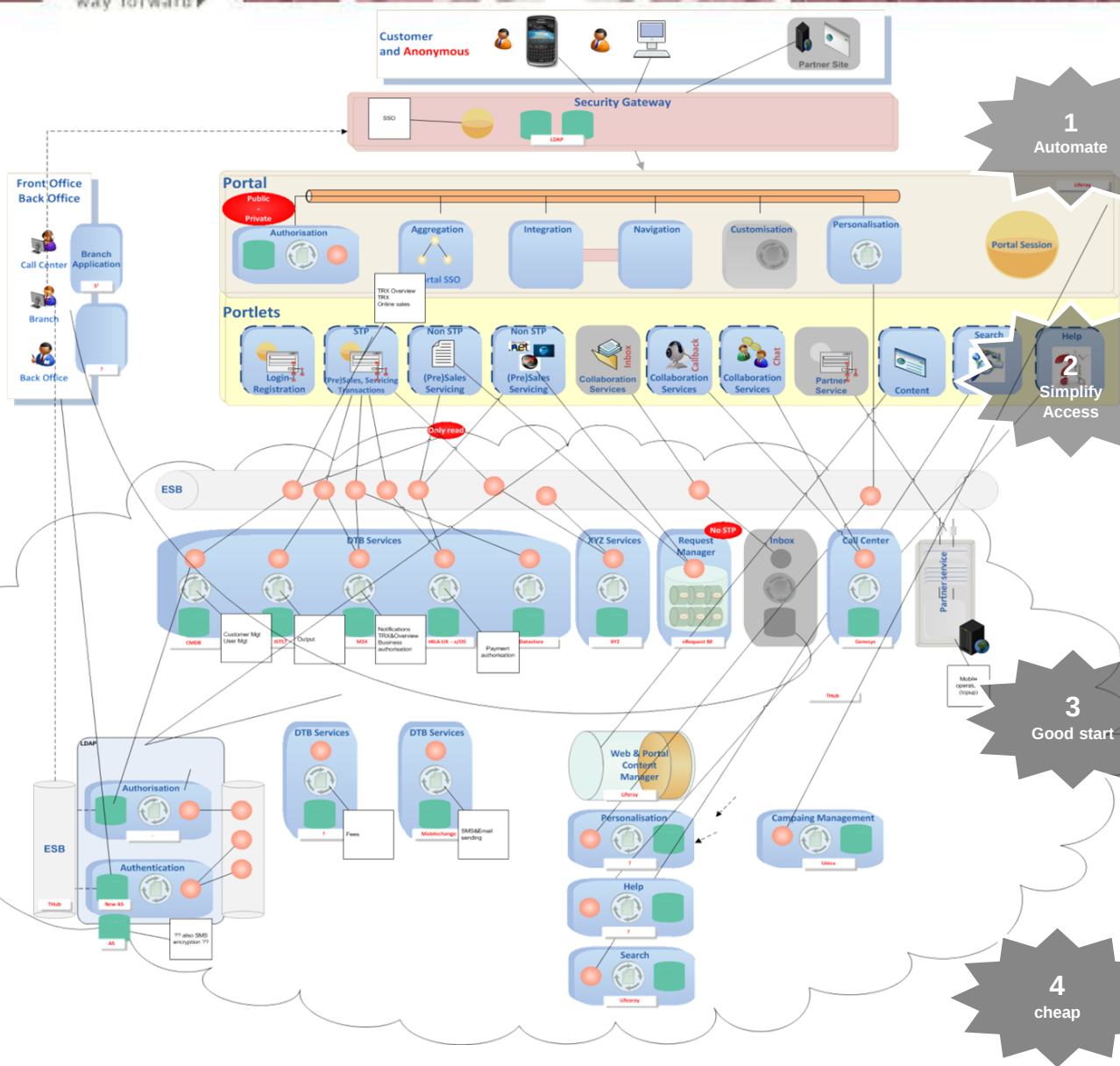
-  Forums
-  Wiki
-  Task Management
-  Chat
-  Blogs
-  Calendars
-  Events
-  Search
-  Notifications
-  Document Library



Application Architecture



Phase 1 Accomplishments



- **Platform & framework** where new services can be plugged-in
- **Unified façade** completely abstracts complexities of organisational business processes and data storage from user's point of view
- **Compartmentalisation** of the existing legacy/spaghetti problem within SOA from the architectural point of view
- Open source **Tactical win** which still fits within the boundaries of long-term strategic goals

1
Automate

2
Simplify
Access

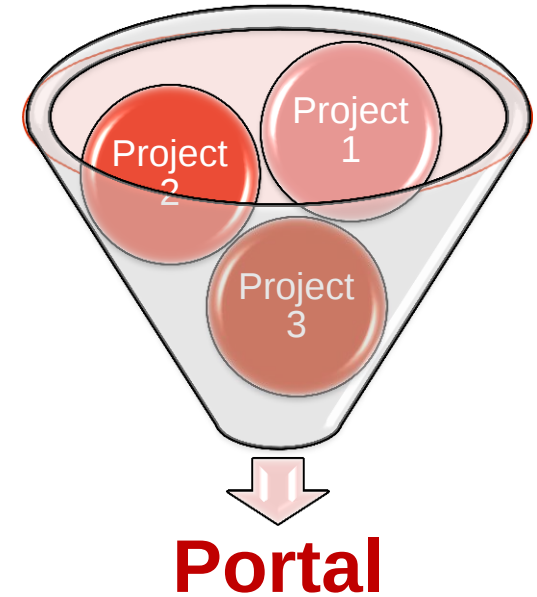
3
Good start

4
cheap

Phase 1

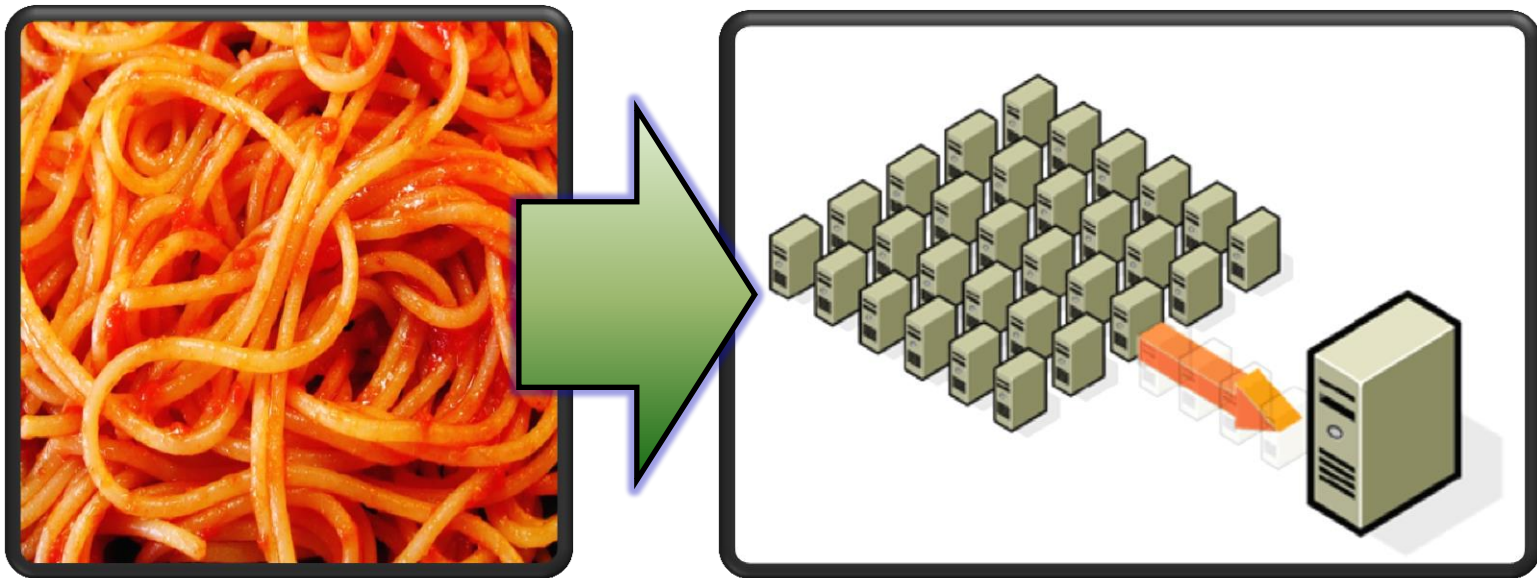
Secondary Accomplishments

- Improvements in application governance
 - Dynamic delivery structure providing increased flexibility for business through “Portlet Factories”
- Full synergy in technologies and frameworks to satisfy business objectives of robust, service-rich portal:



Approach – Phase 2

- Overhaul presentation layer
 - **DONE! (goals 1,2,4)**
- **Consolidate** Keep underlying SOA & Data
 - **Limit effort to this piece alone (goal 3)**
 - Can be done gradually depending on budget & available resource
 - Goal to improve both APEX (through simplifying ease of enhancing) and OPEX (reduction in SM costs)



- De-duplication & consolidation
 - Determine a unique identifier (or set of IDs)
 - Many cases a virtual userId must be used (i.e. 9767865)
 - Tokens which are difficult to remember
 - Users access infrequently
 - Passwords
- Consider integrating with 3rd parties
 - Google apps (OpenID, OAuth 2.0)
 - Facebook (Facebook Connect)

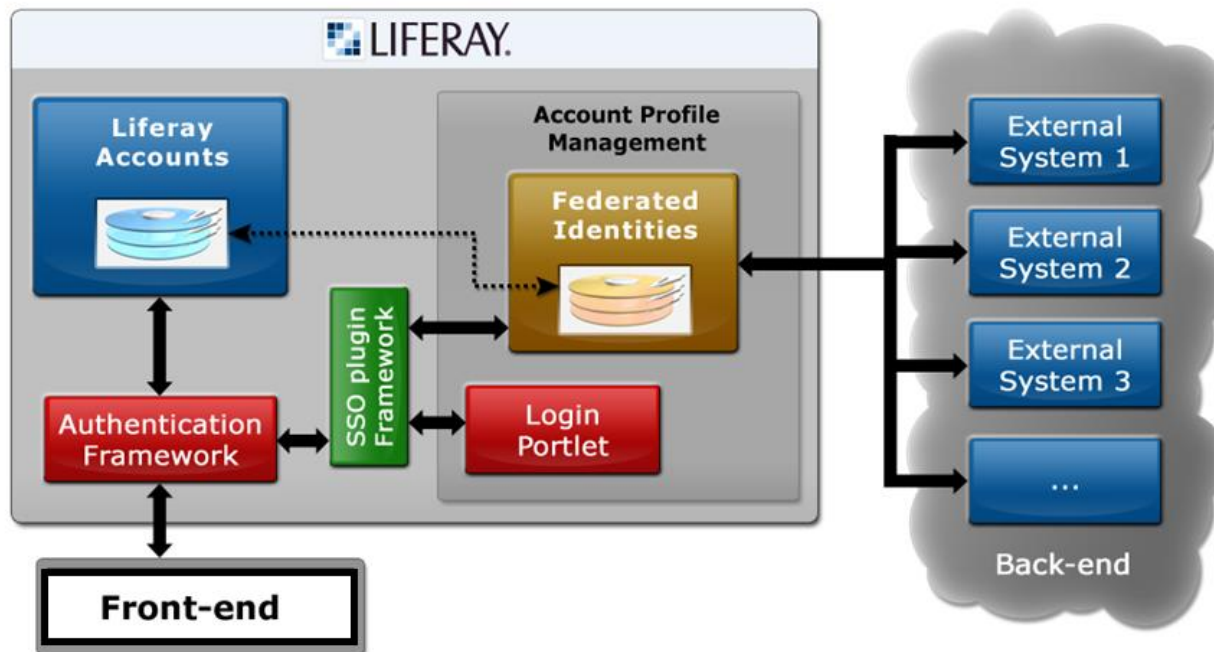


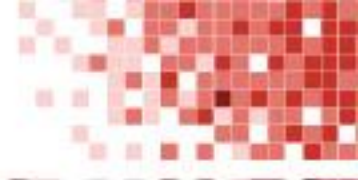
OpenID



Identity Management, SSO, & Liferay

- Aggregate user data based on ID
 - Physically under a single data store
 - Requires a potentially expensive migration (likely in waves)
 - Additional HW & SW (defeats our purpose)
 - Federated identities
 - Stored across multiple distinct identity management systems

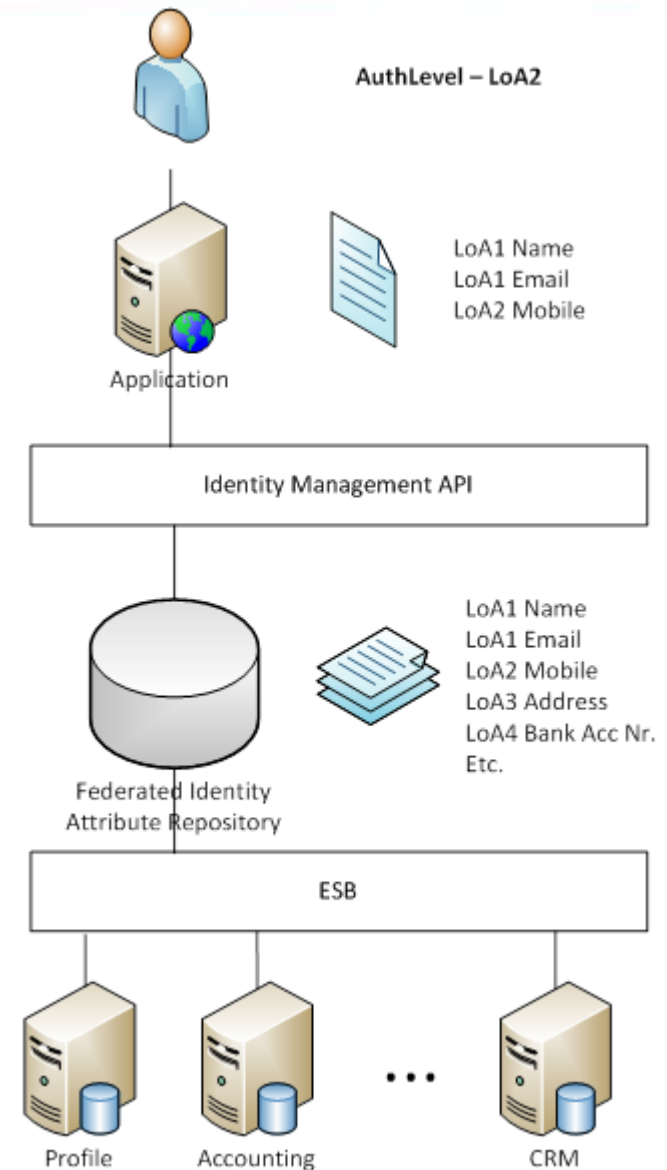




- Different *Levels of Authorisation* (LoA) for different “security zones”
- Increase user convenience by introducing multi-level *authentication*.
 - The more the user *authenticates* themselves, the higher level of *trust* (authorisation/privileges) they are provided.
 - Request further *authentication* for higher security zones on demand (per session)
- Identity Attribute Access control (read / write permissions)
- Reduce the number of user account attributes needed upon registration
 - Extend user profile on demand if required by a higher security zone (e.g. email address, ID information)

Level of Authorisation

- Level of Authorisation (LoA) examples:
 - LoA0: Anonymous
 - LoA1: Minimally Registered
 - LoA2: Registered, verified email
 - LoA3: Verified via SMS
 - LoA4: Account verified against Official Documents
- User *authenticates* with a certain LoA
- Application requests data from Identity Management API
- Federated Identity Management system invokes ESB services and provides a list of attributes available to the user corresponding to his LoA
- Each attributes can have different LoA requirements for different access levels (read vs. write)



Privileges - LoA & ACL

- Examples of basic ACL table for profile attributes:

Attribute	LoA	Privilege
emailAddress	1	READ
emailAddress	2	WRITE
...	...	...
bankAccount	4	READ
bankAccount	4	WRITE

- Identity Management API allows privileges on distributed back office “profile attributes” based on ACL lookups against LoA
- ACL can be applied to more than attributes:
 - Privileges on operations/actions
 - Privileges on Page Views

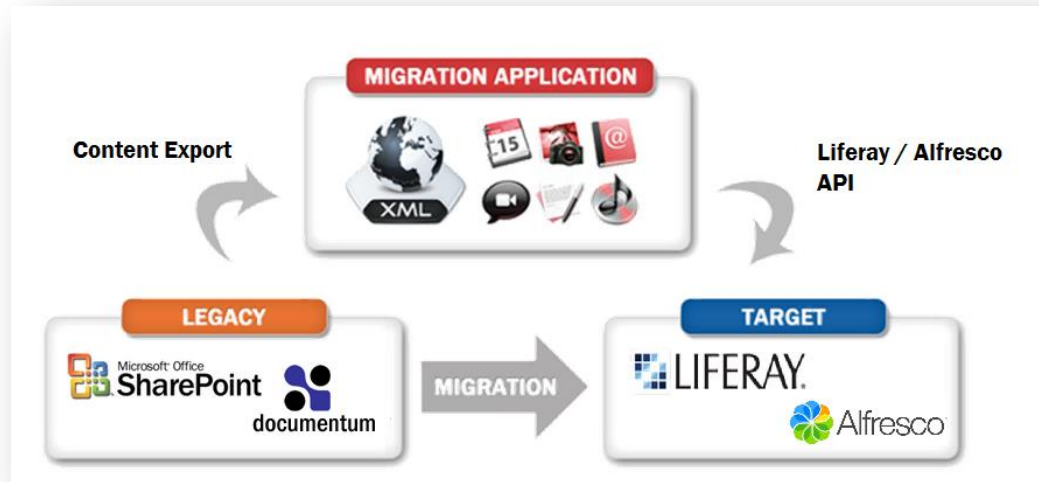
Consolidate CMS(s)

Migrate to open source

Decommissioning portal verticals = decommissioning CMSs

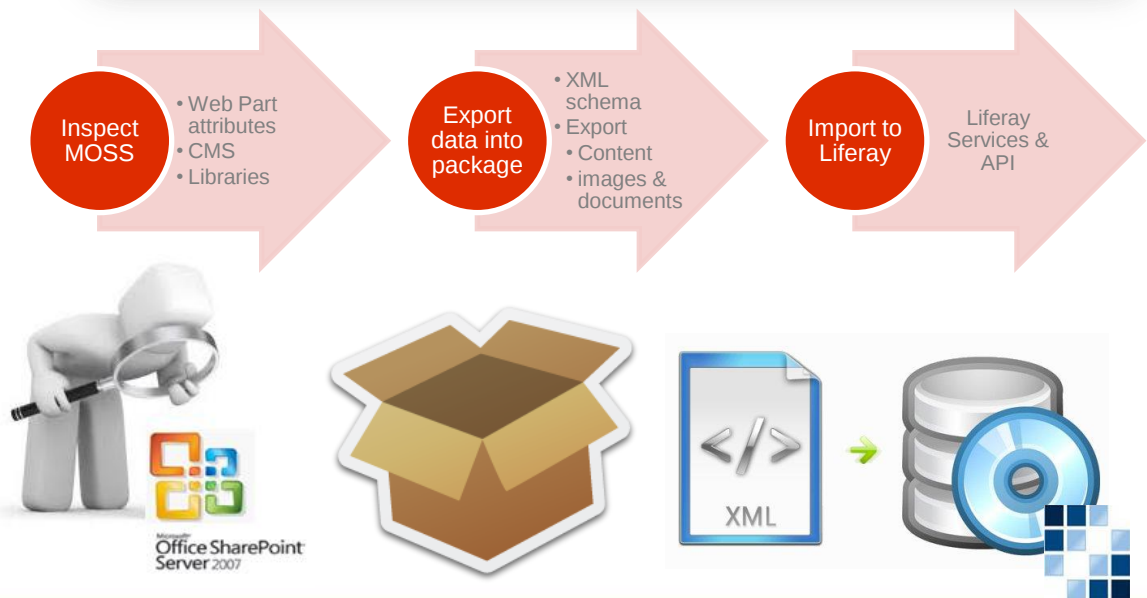
Migration of Content

- Pages & Site Map
- Layouts
- Articles
- Documents & Images
- Look & Feel (theme)



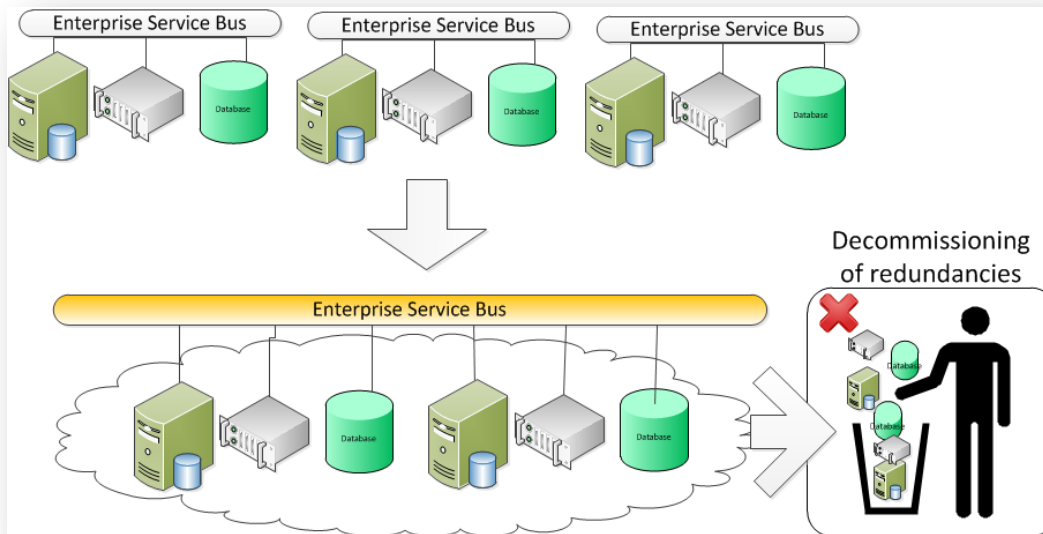
Push-button tool:
Complete migration of
Microsoft SharePoint
into Liferay CMS:

<http://www.liferay.com/video?title=video-web-event-approach-execution-of-migrating-a-microsoft-sharepoint-site-to-liferay>

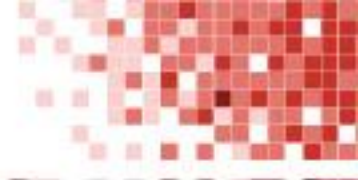


SOA, ESB, and open source

- Adopt a single centralized ESB
 - Orchestration of all business processes



- Consider open source:
 - JBoss Enterprise SOA Platform
- Recommend to move away from SOAP when possible
- Move towards RESTful web services when possible
 - System integration effort considerably lower



- Open source technology has a proven track record in bringing order and cohesion to portal landscapes
- Through Open Source technologies, we can achieve the following business benefits:
 - Stability
 - Lowered cost of maintenance
 - Lowered cost of enhancement
 - Flexibility / vendors
 - Better service - can do more with what we have (in terms of invested effort)

About Emeldi

- Emeldi is an official Liferay service partner in the UK, Czech Republic, & Slovakia
 - e-Commerce / Self-Service / best practice
 - Specialized expert team - Liferay hands-on experience
 - Flexible development process & methodologies
 - Innovation is one of the key differentiators
- 
- Reputable track record delivering solutions to Tier 1 enterprises
 - Competitive value based pricing



Thank you Open for Q&A

Alistair Oldfield – Senior Project Manager

Emeldi Group – Official Liferay Partner
www.emeldi.com

alistair.oldfield@emeldi.com

