Sustainable LIS Augmentation Strategies as Made Possible by Use of Microservices Architecture

Chris Williams, MD
University of Oklahoma Health Sciences Center
Disclosures

In the past 12 months, I have not had any significant financial interest or other relationship with the manufacturers of the products or providers of the services that will be discussed in my presentation.
Learning Objectives

• Learn how to evaluate whether to develop custom LIS functionality
• Define a microservice
• Best practices for implementation
When to Build a Custom Solution
Lessons Learned

1) Ill advised
Lessons Learned

1) Ill advised

1) Fun to watch
Lessons Learned

1) Ill advised

1) Fun to watch

1) Ends the same way, with regret
When to Build a Custom Solution

Define the perceived gap
When to Build a Custom Solution

Define the perceived gap
Is it actually a problem worth solving?
When to Build a Custom Solution

Define the perceived gap
Is it actually a problem worth solving?

Costs: Hardware, Development cost, Maintenance
Benefits: Reduced FTE, Increased Quality, etc.
Opportunity costs: ???
When to Build a Custom Solution

Build vs Buy
Failing to Plan is Planning to Fail

Develop a Project Charter

1. Timeline
2. Scope
3. Deliverables
4. How will you measure success
Microservices
Traditional Software Development

- Large, monolithic applications
- Object oriented design
- Shared data model, state, etc.
Applications composed of modules:
• Perform a function or related functions
• Loose coupling
• Maintainable/Upgradable
Technical Note

The growing need for microservices in bioinformatics

Christopher L. Williams¹, Jeffrey C. Sica¹, Robert T. Killen¹, Ulysses G. J. Balis¹

¹Department of Pathology, Division of Informatics, University of Michigan, Ann Arbor, MI, USA

E-mail: *Dr. Ulysses G. J. Balis - ulysses@med.umich.edu
*Corresponding author

Received: 16 January 2016          Accepted: 16 August 2016          Published: 29 November 2016

Abstract

Objective: Within the information technology (IT) industry, best practices and standards are constantly evolving and being refined. In contrast, computer technology utilized within the healthcare industry often evolves at a glacial pace, with reduced opportunities for justified innovation. Although the use of timely technology refreshes within an enterprise’s overall technology stack can be costly, thoughtful adoption of select technologies with a
Microservices

Extension of Services Oriented Architecture:

- Well-defined (narrowish) scope
- Web based interfaced
- Well defined API
  - Independent
Microservices

Many software companies have embraced microservices:

- Parallel development
- Use up-to-date tools without affecting legacy projects
- Maintainability
- Agile development
- etc...
Containerization
Containers

Analogous to a lightweight VM

• Encapsulates an application and dependencies
• Easy to transport between different environments
• Easy to scale
Container Orchestration

Microservices

kubernetes

RANCHER
Container Orchestration

Acts as an automated backend server administrator

- Manages compute resources
- Performs health checks on services
- Load balancing
- Detects and attempts to fix errors
- Misc
  - Logging, secrets, storage, etc.
Microservices

Your microservices should be thought of as cattle rather than pets
Deployment
Security Basics

Authentication
Authorization
Encryption
Logging/Auditing
Work with Central IT

Expect pushback
• You have strong justification for this misadventure
Leverage as much as you can
• VMs, active directory, database support, logging, etc.
Have patience and educate
• Find a friendly liaison
Key points

- DO NOT design a mini-LIS
- DO NOT engineer job security
- DO prioritize maintenance and stability over development, play the long game
- Do use central IT resources as much as possible
Failure Mode

2 options:
1. Catastrophic Failure - default
2. Failing Gracefully
PathTrack
IHC Order Form

IF YOU COULD FAX IHC REQUESTS

THAT WOULD BE GREAT
<table>
<thead>
<tr>
<th>IHC Orders</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CD10</td>
<td>CD117 (c-kit)</td>
<td>CD138</td>
<td>CD15 (Leu M-1)</td>
<td></td>
</tr>
<tr>
<td>CD20 (L26)</td>
<td>CD3</td>
<td>CD30 (Ber H2)</td>
<td>CD14</td>
<td></td>
</tr>
<tr>
<td>CD4</td>
<td>CD45 (LCA)</td>
<td>CD45RO (UCHL-1)</td>
<td>CD5</td>
<td></td>
</tr>
<tr>
<td>CD56 (NCAM)</td>
<td>CD79a</td>
<td>CD8</td>
<td>Ewings (CD99)</td>
<td></td>
</tr>
<tr>
<td>NSE</td>
<td>PAX8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Manifesto

- I will use LIS functionality when possible
- I will not create a source of truth
- I will not create a shadow database
References

1. docker.com
2. kubernetes.io
Questions