The Ah Ha Moment: How eXp Realty Reached The Decision to Build a Microservices Architecture.

Steve Ledwith / VP Engineering / eXp Realty, LLC
Agenda

1. Introductions & Context
2. Enterprise Growth
3. Challenges
4. Research
5. The Solution
6. Enterprise Platform
About eXp Realty, LLC

Cloud-based Real Estate Brokerage

• Largest real estate brokerage by geography in North America.

• Only brokerage to operate as a company-owned organization in all 50 U.S. States.

One of the Fastest Growing Firms

• More than 18,000 agents across

  • 50 U.S. States
  • District of Columbia
  • Three Canadian Provinces
  • Publicly traded on NASDAQ: EXPI
About Steve Ledwith

Vice President, Engineering

- More than 20 years experience in developing technology for the real estate industry

- Responsible for planning and management of the company’s engineering strategy, and for developing and maintaining a technical roadmap that will continue to deliver the most innovative products and services for the eXp Realty agents, brokers and staff.

- Figure head of an amazing, talented, innovative team.
eXp Enterprise – Then and Now

Look how far we’ve come!

• eXp Enterprise is one application, running in the Mendix Cloud.
## eXp Enterprise – Metrics

<table>
<thead>
<tr>
<th>App Container</th>
<th>Strato</th>
<th>Meso (+)</th>
<th>Iono (++)</th>
<th>Magneto (+++)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mx RAM</td>
<td>2 GB</td>
<td>2 GB</td>
<td>8 GB</td>
<td>16 GB</td>
</tr>
<tr>
<td>DB CPU</td>
<td>0,5 core</td>
<td>2 core</td>
<td>2 core</td>
<td>4 cores</td>
</tr>
<tr>
<td>DB RAM</td>
<td>1 GB</td>
<td>8 GB</td>
<td>8 GB</td>
<td>16 GB</td>
</tr>
<tr>
<td>DB Storage</td>
<td>5 GB</td>
<td>20 GB</td>
<td>20 GB</td>
<td>80 GB</td>
</tr>
<tr>
<td>File storage</td>
<td>20 GB</td>
<td>20 GB</td>
<td>20 GB</td>
<td>320 GB</td>
</tr>
<tr>
<td>Backup</td>
<td>25 GB</td>
<td>40 GB</td>
<td>100 GB</td>
<td>400 GB</td>
</tr>
<tr>
<td>Network</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>Unlimited</td>
</tr>
<tr>
<td>Encryption at rest</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
## eXp Enterprise – Metrics

<table>
<thead>
<tr>
<th>App Container</th>
<th>Strato</th>
<th>Meso</th>
<th>Iono(++)</th>
<th>Magneto(++)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIX RAM</td>
<td>2 GB</td>
<td>2 GB</td>
<td>8 GB</td>
<td>16 GB</td>
</tr>
<tr>
<td>DB CPU</td>
<td>0.5 Core</td>
<td>2 Core</td>
<td>2 Core</td>
<td>4 Cores</td>
</tr>
<tr>
<td>DB RAM</td>
<td>1 GB</td>
<td>8 GB</td>
<td>8 GB</td>
<td>16 GB</td>
</tr>
<tr>
<td>DB Storage</td>
<td>5 GB</td>
<td>20 GB</td>
<td>20 GB</td>
<td>80 GB</td>
</tr>
<tr>
<td>File Storage</td>
<td>20 GB</td>
<td>20 GB</td>
<td>20 GB</td>
<td>320 GB</td>
</tr>
<tr>
<td>Backup</td>
<td>25 GB</td>
<td>40 GB</td>
<td>100 GB</td>
<td>400 GB</td>
</tr>
<tr>
<td>Network</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>Unlimited</td>
</tr>
<tr>
<td>Encryption at rest</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Thanos (WTF)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16 GB</td>
<td>16 Cores</td>
<td>64 GB</td>
</tr>
<tr>
<td></td>
<td>100 GB</td>
<td>320 GB</td>
<td>400 GB</td>
</tr>
<tr>
<td></td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>Unlimited</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Challenges – Hosting Resources

Mendix Cloud v3

- As shown earlier, we use a tremendous amount of resource in the Mendix Cloud.

- In June 2018, we realized the V3 Cloud would not be able to support us.

- We were already running on an XXL Container, and there was no more room.

- We migrated our 6.10 application to the V4 Cloud

- (Story for another time!)
Challenges – Technical Debt

Best Practices are only a Suggestion! Right?

• Our team grew at an impressive rate, as did the feature requests

• We built a number of new tools on top of functionality not designed to support multiple workflows

• Input from users was limited, and we tightly coupled a lot of business logic across modules
Challenges – Entity Entanglement

Agent Database Domain Model*
Challenges – Pushing the Limits

Mendix CloudFoundry Upgrade

• As we’ve grown, we’ve pushed the limits of the Mendix Cloud. Following the Cloud Foundry upgrade, we ran into performance issues, and frequent application crashes.

• The root cause was a new limit in Cloud Foundry which didn’t work for our application.

“Note that > 1000 threads is indeed way beyond what is considered reasonable on the Mendix Platform, this is why it was not caught in testing. eXp needs to consider Refactoring their Application to bring this number down.”

–Mendix Cloud Support
Realization & Research

Seminal Moment
In January 2018 we realized our current path forward wasn’t going to work.

With three distinct teams working on Enterprise, we were constantly running into issues with scheduling, timing, and delivery.

Research Initiative
Spent a considerable amount of time reading about how leading technology and software companies were enabling their engineers and teams to accomplish amazing things.
Research – Throw Hardware at It!

- Most straight forward approach!
- Add resources to support our growth, and keep the application running
- Buy time to make better decisions

Challenges

- Only so much time you can buy
- Limits of the Mendix Cloud - containers only get so big
- Couldn’t meet our growth, and does not scale
Research – Rewrite, Refactor, Optimize

- Would keep our business logic in place
- Correct some mistakes we’ve made
- Handle roles much better!

Challenges

- Would have two applications in process for a long time during the rewrite
- New features would have to be in both places
- Still have one giant app, with prospect of thousands of users
Research – Hub & Spoke

• Centralized integrations
• One main system to handle interaction with each satellite system
• Easy to understand

Challenges

• Similar to our current situation
• One main app which would require constant updates to allow for new integrations
• One single point of failure
Research – Microservices

• Right sized, purpose built, process oriented applications
• Independent development and deployment schedules
• Autonomy and Flexibility

Challenges

• New process and style, requires a different way of thinking
• Not how we’ve always done it
• Limited understanding across the teams
• New versions of existing code - like rewriting
Microservices - It’s got a Ring to It!

Microservices is a software development technique — a variant of the service-oriented architecture (SOA) architectural style that structures an application as a collection of loosely coupled services. In a microservices architecture, services are fine-grained and the protocols are lightweight.
Microservices - It’s got a Ring to It!

“While there is no precise definition of this architectural style, there are certain common characteristics around organization around business capability, automated deployment, intelligence in the endpoints, and decentralized control of languages and data.”

- Martin Fowler

https://martinfowler.com/articles/microservices.html
The Enterprise Platform

Finding a Solution was only the Start

• Alignment
  • Business
  • Engineering
  • Product
• Socialization
  • Teaching
  • Training
• Experiments
  • Roll-out
  • Innovation Teams

“Delivering business Value requires input, Logic, and feedback, With a constant focus on what the business truly needs.”
Microservices - Architecture

1. Strategic Goals
   • Enable Business at Scale
   • Support Business Innovation
   • Support Third Party Integration

2. Architectural Principles
   • Reduce Complexity
   • Favor Rapid Development and Feedback
   • Consistent interfaces and right tool for the job

3. Design & Delivery
   • HTTP / REST / OData
   • Continuous Deployment and Integration
   • Limit shared code between projects
   • Small standalone services
Innovation Teams – Deliver Value

Strategic Mindset – enable us to accomplish new things. Stand out in the marketplace.

Revenue Focus – add new revenue streams, reduce churn, and increase productivity.

Cost Reduction – Automate tasks, eliminate costs, reduce work.
The Enterprise Mindset

Low code doesn't mean "no code." Anyone can build an application in Python, C#, Mendix, or Java. Delivery of enterprise-class software with real business values requires planning, talent, input, logic, feedback, and careful execution with a constant focus on what the business truly needs.
Questions
Resources – Build Your Knowledge

Why DevOps Will Fail Without Microservices


The Microservices You Need for DevOps

- https://www.mendix.com/blog/the-microservices-you-need-for-devops/

How Microservices and DevOps Help CIOs Realize Business-IT Alignment


Autonomous Units are Critical for Microservices

- https://www.mendix.com/blog/autonomous-is-critical-for-microservices/
Resources – Build Your Knowledge

Building Microservices by Sam Newman


Production Ready Microservices by Susan Fowler

• https://www.amazon.com/Production-Ready-Microservices-Standardized-Engineering-Organization/dp/1491965975

Mendix Microservice Webinar