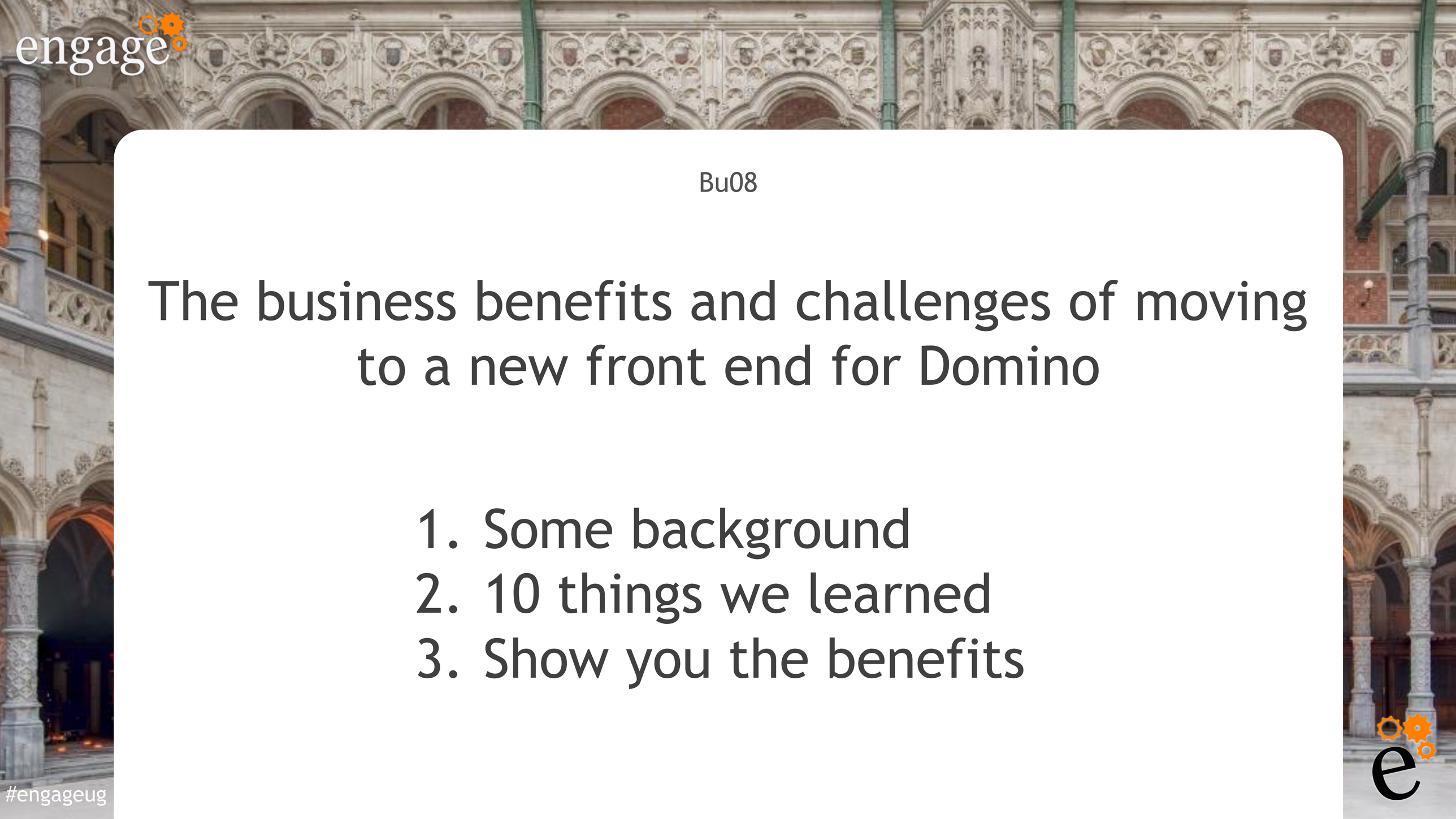
Bu08

DRAFT

The business benefits and challenges of moving to a new front end for Domino

Sean Cull

David Harding



Bu08

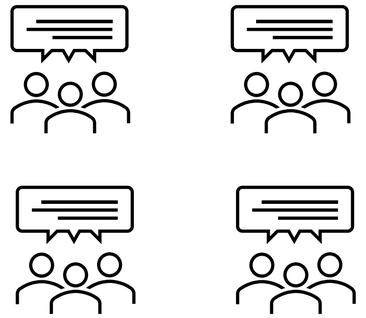
The business benefits and challenges of moving to a new front end for Domino

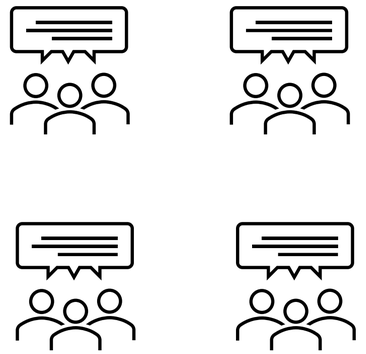
1. Some background
2. 10 things we learned
3. Show you the benefits

About FoCul

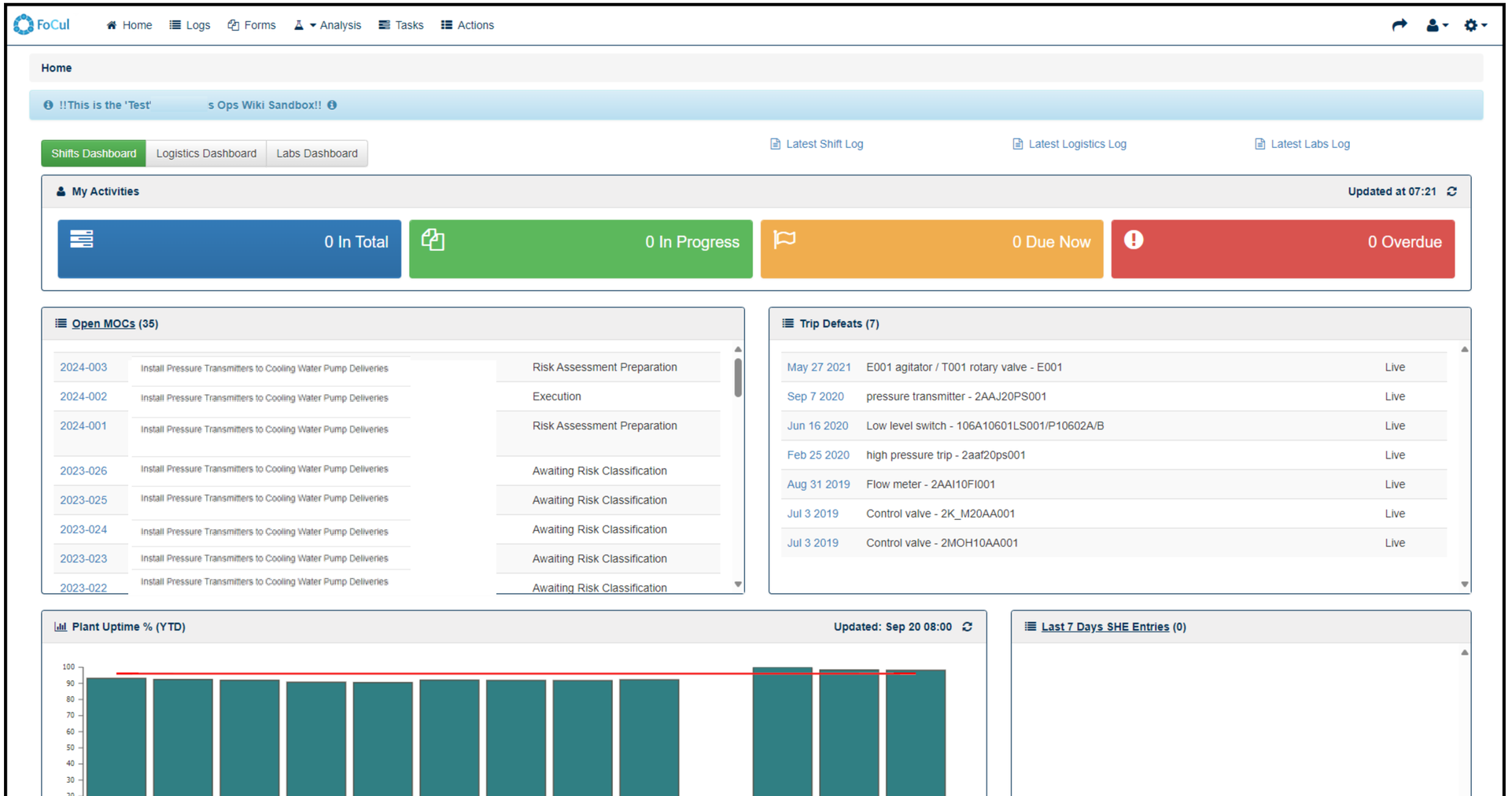
HCL Notes
Xpages

Customers





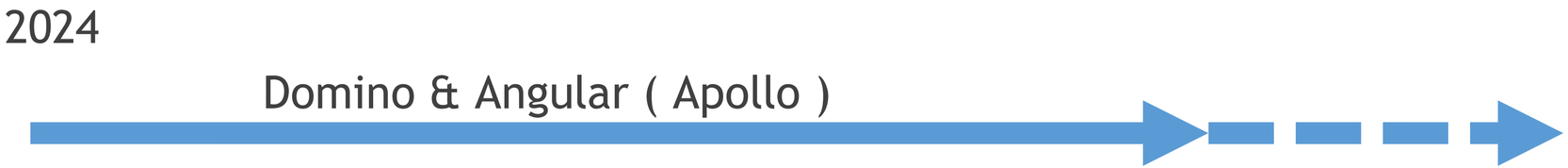
Niche SaaS Solutions



Timeline

2000
2008
2010
Lotus Notes

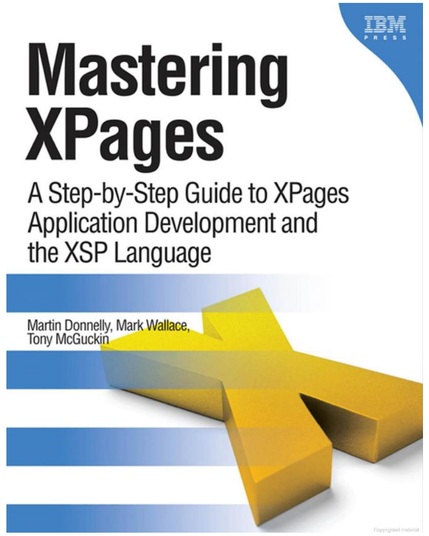
2010
2019
2024
Domino XPages (Mosaic 5th Gen)


2024
Domino & Angular (Apollo)

2009 XPage Journey

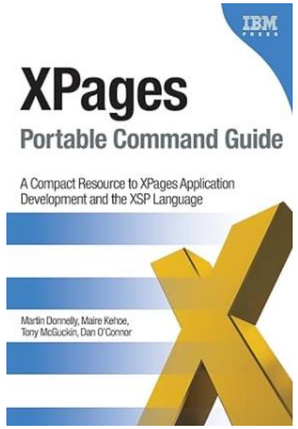
Significant IBM Enablement

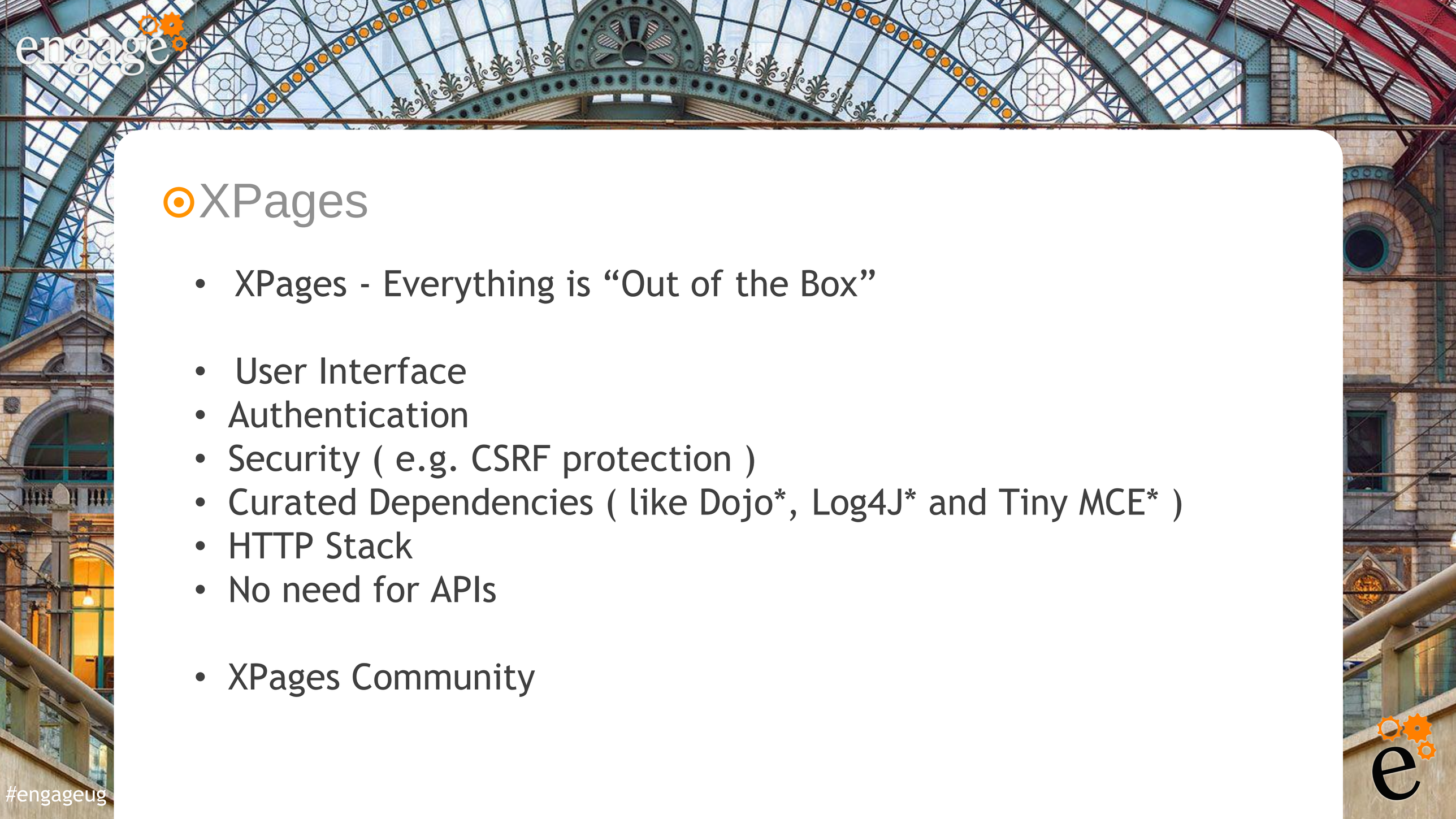
- Matt White (enabled by Tim Clark – IBM, now HCL)
- Declan Lynch
- Martin Donnelly - IBM
- Tony McGuckin – IBM
- Paul Withers – Intec (now HCL)
- And many more



Developer Journey

- Went from “Line of Business developers” to Developers
  - Java
  - Object Oriented approach
  - Source Control



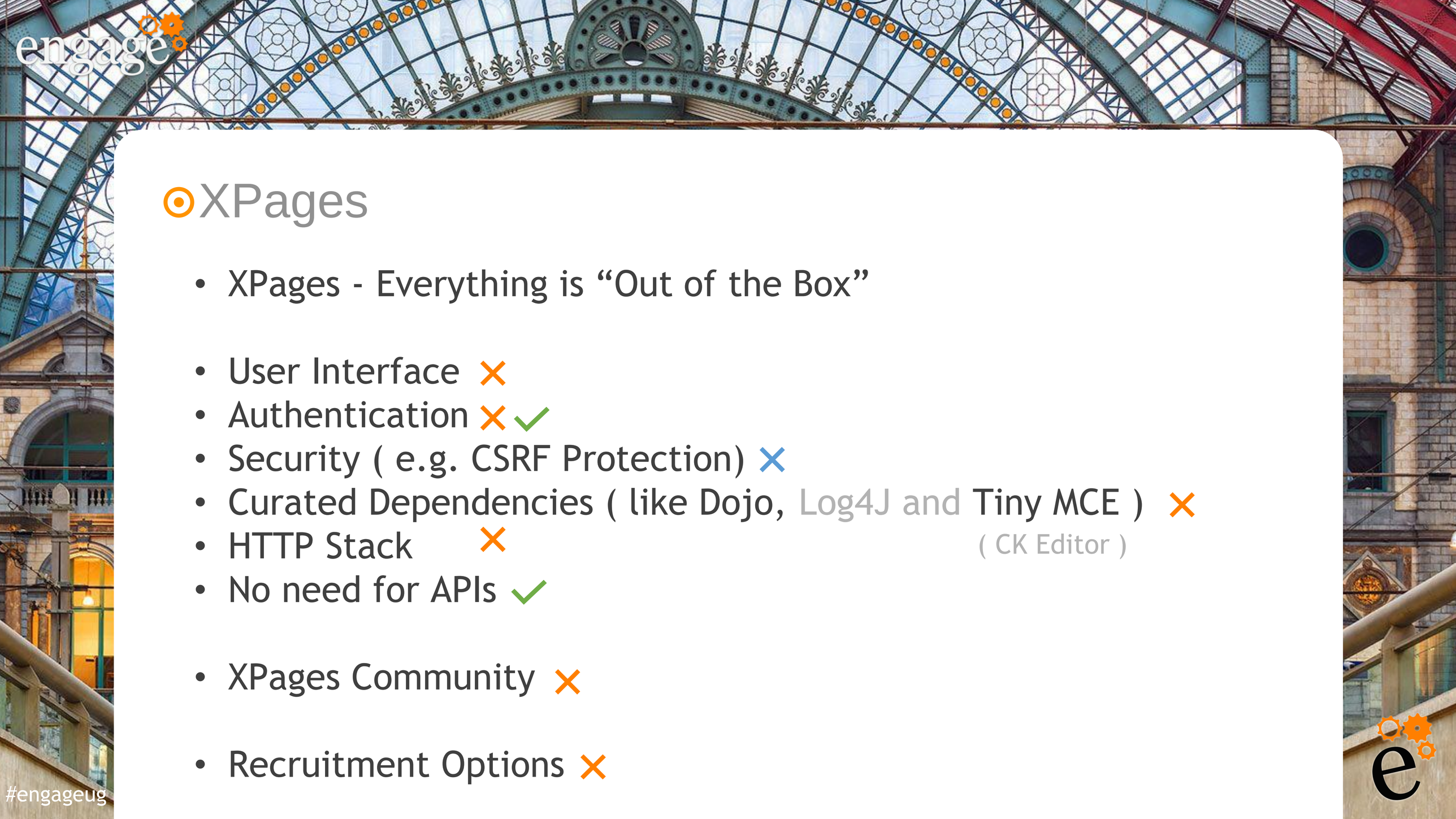


XPages

- XPages - Everything is “Out of the Box”

- User Interface
- Authentication
- Security (e.g. CSRF protection)
- Curated Dependencies (like Dojo*, Log4J* and Tiny MCE*)
- HTTP Stack
- No need for APIs

- XPages Community



XPages

- XPages - Everything is “Out of the Box”

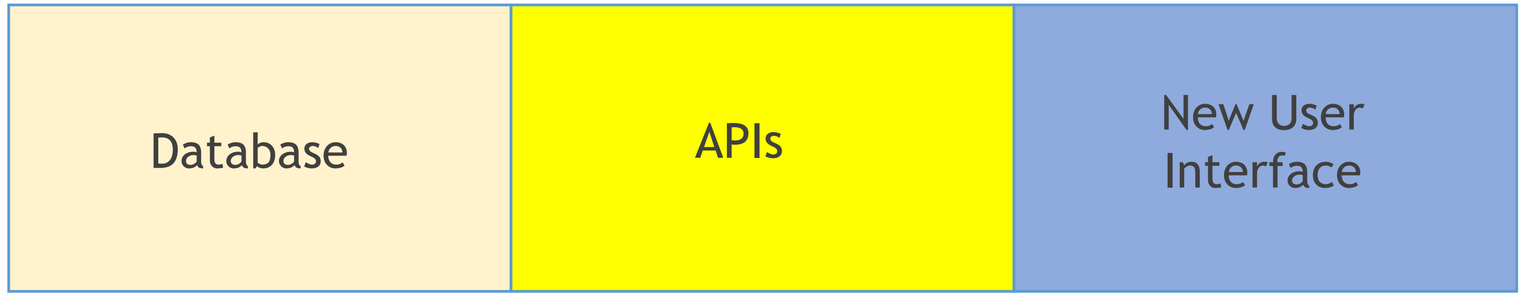
- User Interface ✗
- Authentication ✗ ✓
- Security (e.g. CSRF Protection) ✗
- Curated Dependencies (like Dojo, Log4J and Tiny MCE) ✗
  (CK Editor)
- HTTP Stack ✗
- No need for APIs ✓

- XPages Community ✗

- Recruitment Options ✗

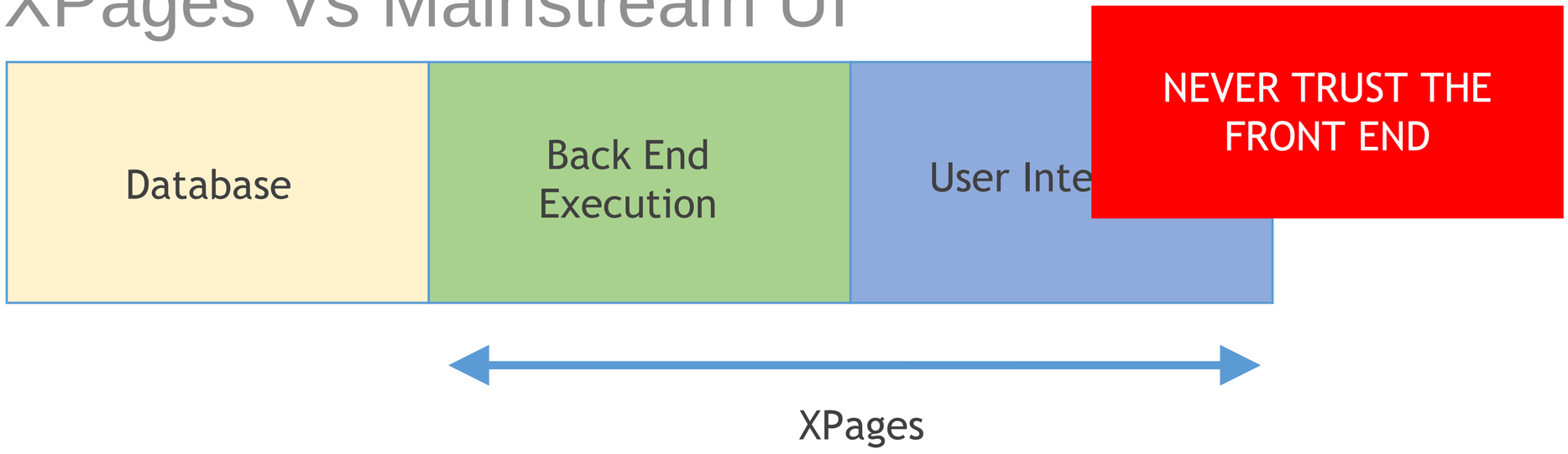
XPages Vs Mainstream UI

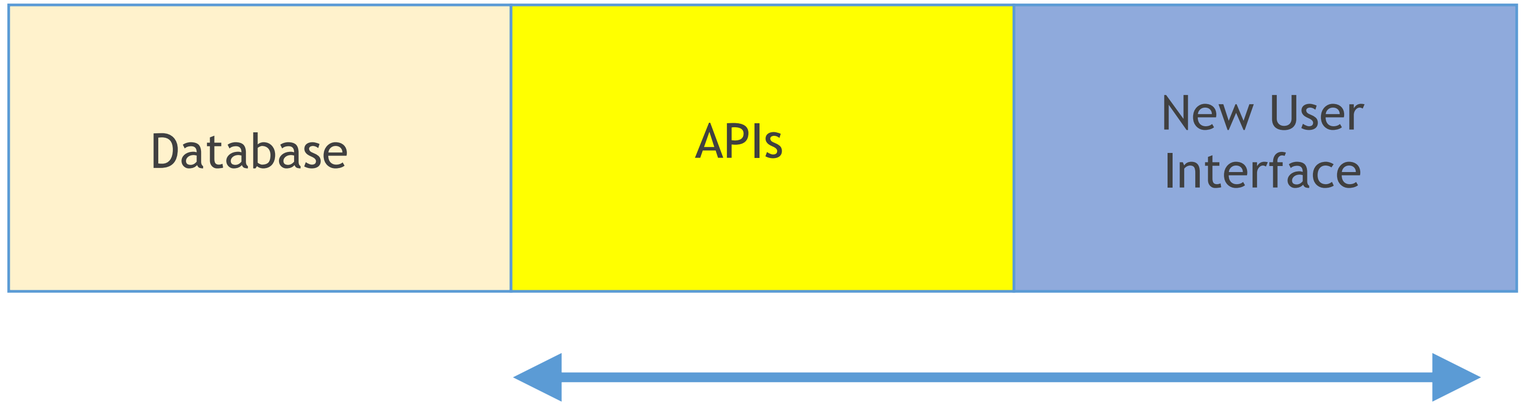
XPages



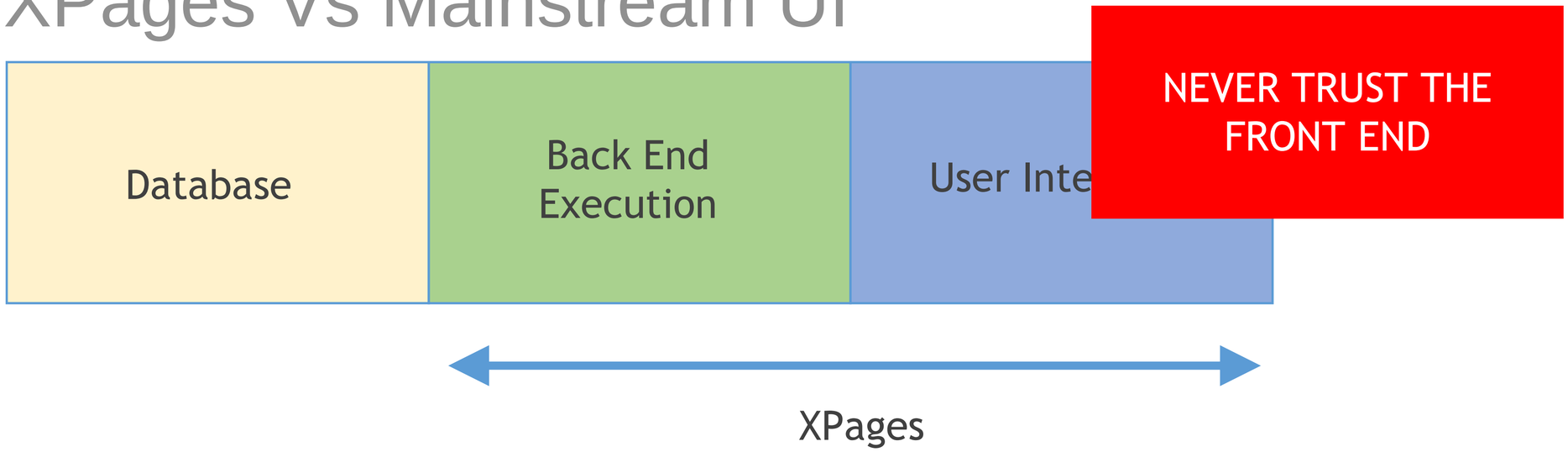
Management's theory on XPage replacement

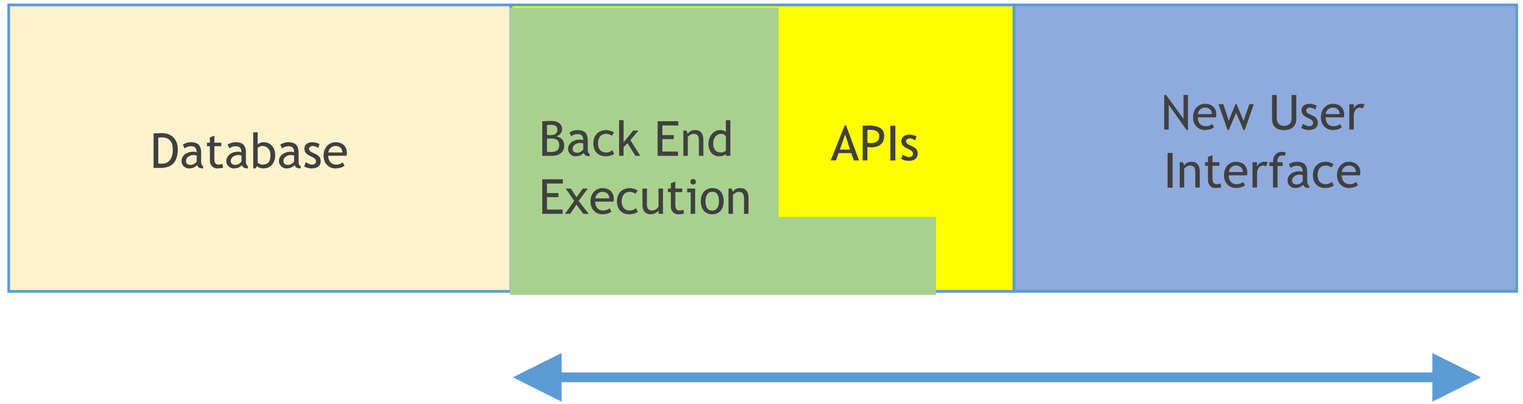
XPages Vs Mainstream UI





XPages Vs Mainstream UI



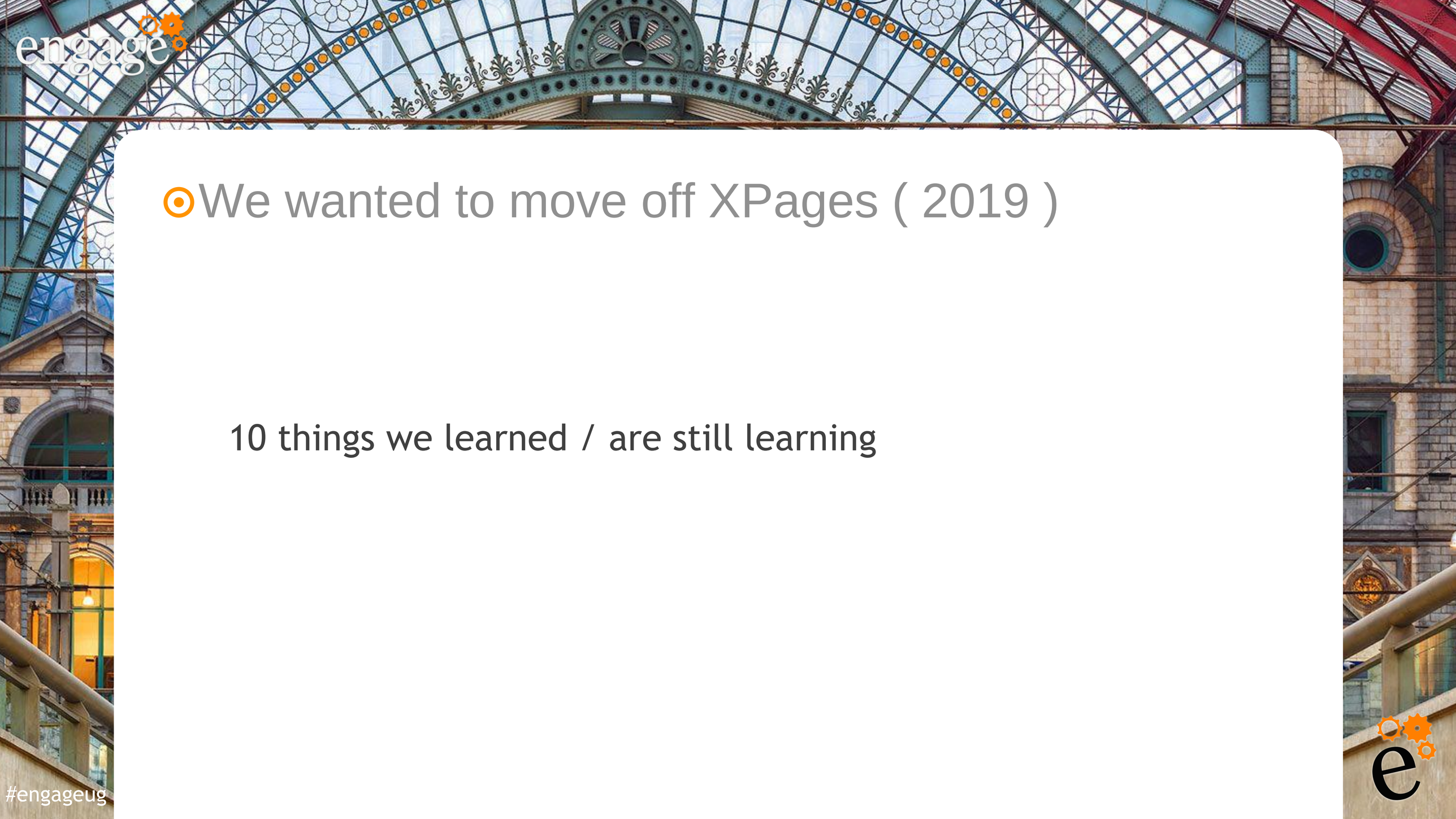


- We wanted to move off XPages (2019)

so……..

We wanted to move off XPages (2019)

- No HCL APIs - pre Domino Rest APIs (DRAPI)/Project Keep - but would probably take same route same again
- Heavy investment in JAVA Business Code
- Small Team
- Much reduced Domino community support compared to 2010



- We wanted to move off XPages (2019)

10 things we learned / are still learning

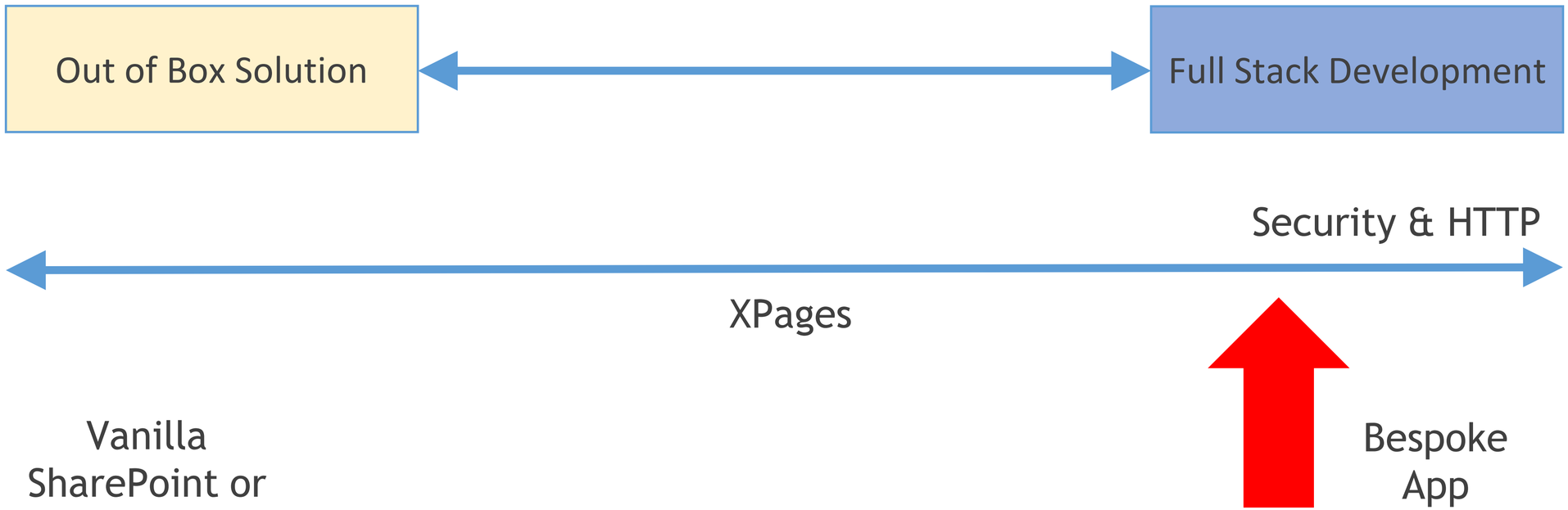
1. What do you want to end up with ?

XPages

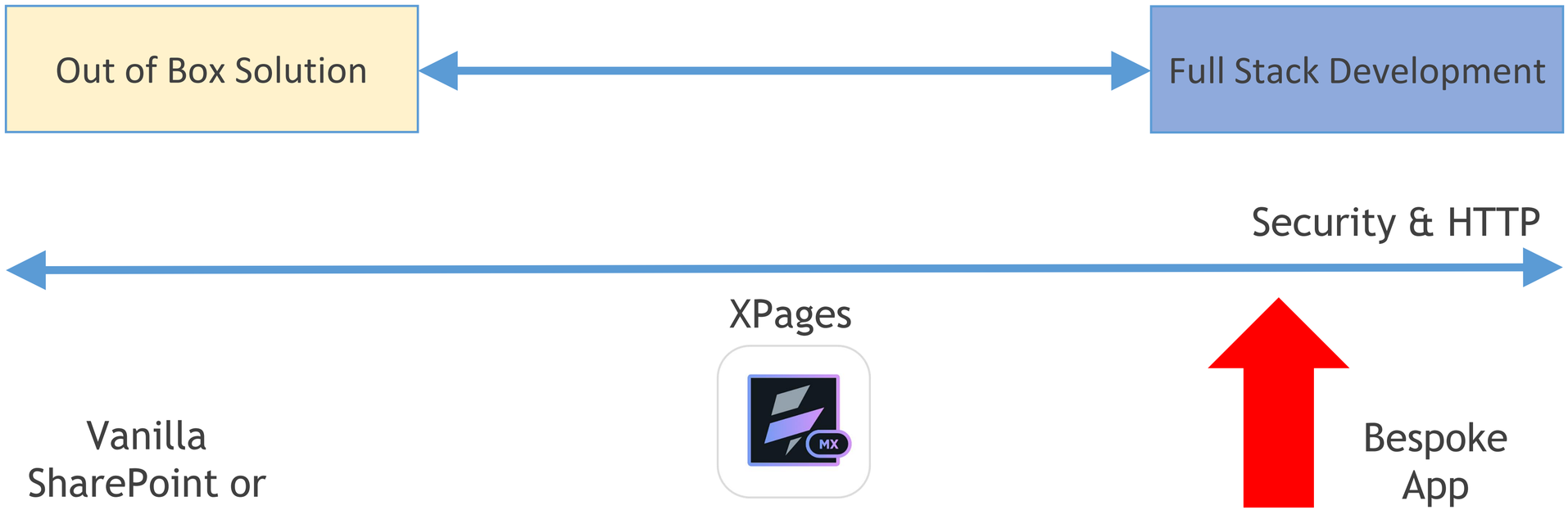
Vanilla
SharePoint or
Connections

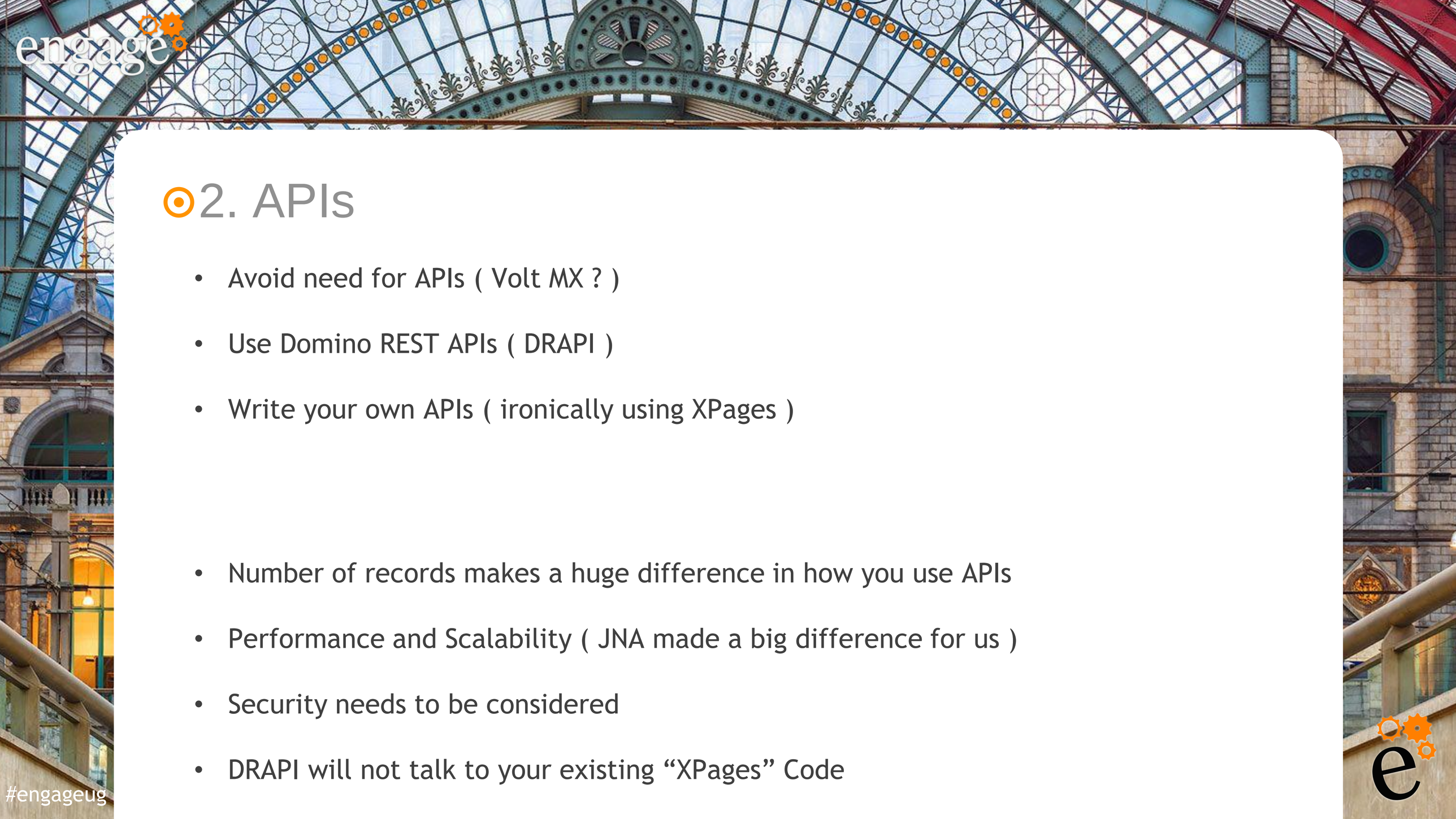
Bespoke
App

1. What do you want to end up with ?



1. What do you want to end up with ?





2. APIs

- Avoid need for APIs (Volt MX ?)
- Use Domino REST APIs (DRAPI)
- Write your own APIs (ironically using XPages)

- Number of records makes a huge difference in how you use APIs
- Performance and Scalability (JNA made a big difference for us)
- Security needs to be considered
- DRAPI will not talk to your existing “XPages” Code

3. Front End Frameworks

- React / Next.js
- Angular
- Vue.js
- ….others

- Number of records makes a huge difference in how you use frameworks
- Initially we thought we would go React but we went with Angular

- Why Angular ?
- Long term outlook – “The Lotus notes of web frameworks” Sean Cull
- Opinionated – better for teams who are learning ?
- Typescript – more structured ?
- Fewer Dependencies – glad we realised
- Some Domino & Angular enthusiasts (but not many actual code examples)

3b. Front End Frameworks+proprietary components?

- We opted to go “Angular Material” – not 100% sure this was right.

- Syncfusion
- Telerik
- Ionic (+ PWAs)
- (Bootstrap)

- Surprised this list isn’t bigger
- There are also “back end” frameworks like Express

4. Knowledgeable Framework Developer

- You don’t know what you don’t know
- In an ideal world you would have someone who knew Domino and your preferred front end – We did ☺
- Chicken and Egg – before or after framework selection ?
- This recruitment will shape your architecture for 5+ years
- Need to quickly establish a good shared understanding of the business objectives e.g.
  - How tied are you to the current look and feel ?
  - Where does mobile fit ?
  - How will documentation be done, and by whom ?

5. Compute Layer

- We have a heavy investment in Java
- Domino Rest APIs (DRAPI / Project KEEP) cannot easily communicate with XPages code
- LotusScript has some new HTTP classes but Java feels more “Enterprise”
- Realistically if you are staying with Domino you need to use Java*
- If you are not going to stay Domino there are more options, including Java.

* Opinionated management view – could be wrong

6. Security / Authentication

- Domino authentication options are poor but are improving with 14+
- For single tenant deployments SSO is becoming the norm
- Think about MFA – Domino MFA may not suit
- Think about CSRF protection
- Remember your API calls need to handle time outs, server restarts etc.

- This topic causes us the most angst
- This all came out of the box with XPages (although CSRF protection isn't great)

OWASP Top Ten 2021 : Related Cheat Sheets

7. Project Processes

- Good opportunity to improve your documentation **BEFORE** you start
- Your team needs to be using Source Control
- You need to work on an MVP* basis
- Multiply your resource expectations by a factor of 4+

7b. Project Strategy – **M**inimum **V**iable **P**roduct

- “The simplest application that you can still be proud of”
- There are HUGE opportunities for scope creep with modernisation
  - Lots of shiny new toys
- You ~~want~~ need to give your existing customers shiny toys to make them enthusiastic about moving to the “new” platform
- You are doubling the skill dependencies (risks) in your business
- Your tech team needs to learn to walk before they can run e.g. testing, debugging

7c. Project Strategy – **M**inimum **V**iable **P**roduct

- You need to make the new technology tangible

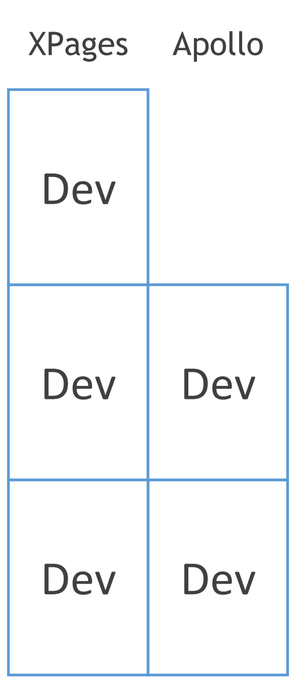
- But need to be careful setting expectations on timescales too

  Hybrid Applications ? (Apollo Lite)
- Administrator Functionality
- Dashboards
- iFrames for XPage content (Apollo Lite)
- Or initially only new customers on new platform (tricky)

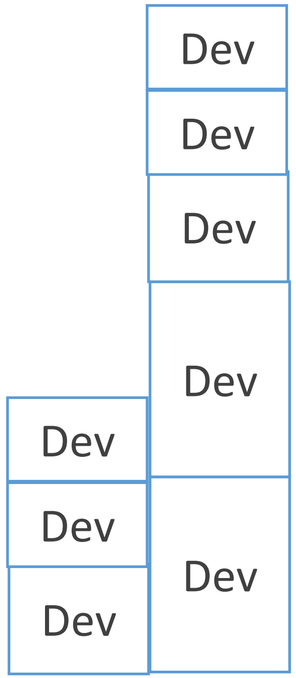
8. Short term resourcing

100% Costs
100% throughput

125% Costs
75% throughput



125% Costs
37% throughput



125% Costs
90% throughput+*

8b. Longer Term Resourcing

- Easier to Recruit

9. Cross Skilling

- Good Angular Training Resources (e.g. Udemy)
- Remember Typescript
- Remember Node / NPM / Dev Tools
- Remember Domino Admin
- Remember Automated Testing*
- Remember Continuous Integration*

- Daunting Task

10. Infrastructure

- We serve Angular from Domino
- Caching (browser and application) takes on a whole new relevance
- HTTP Redirect Rules are important

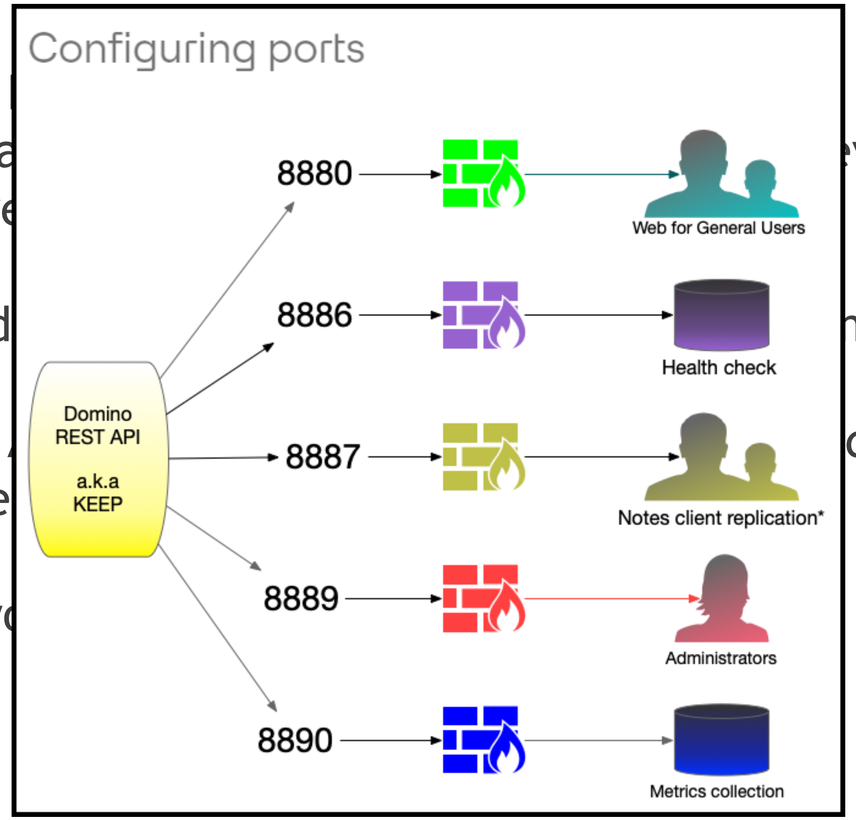
- We can now do live code updates without customer down time

- If you use Domino Rest APIs (DRAPI) you will need either a dedicated API server or a proxy server to combine port 443 traffic*

- Moving to Containers would mean more significant change

10. Infrastructure

- We serve Angular from
- Caching (browser and a
- HTTP Redirect Rules are
- We can now do live cod
- If you use Domino Rest
  API server or a proxy se
- Moving to Containers w



10. Infrastructure

- We serve Angular from Domino
- Caching (browser and application) takes on a whole new relevance
- HTTP Redirect Rules are important

- We can now do live coded updates without customer down time

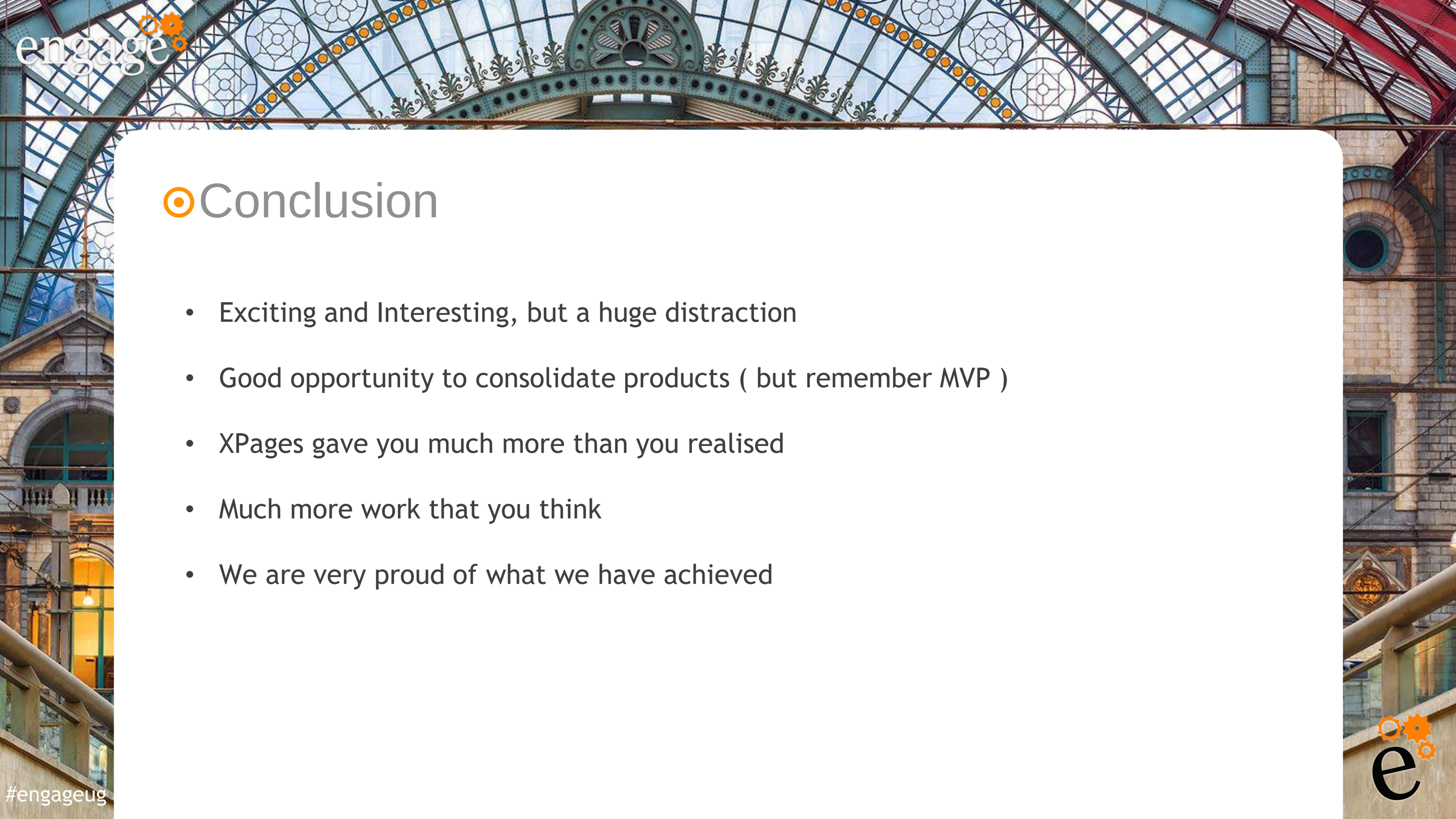
- If you use Domino Rest APIs (DRAPI)you will need either a dedicated API server or a proxy server to combine port 443 traffic*

- Moving to Containers would mean more significant change

What we actually did (are doing)

- Apollo Lite
- Apollo

engage
DEMO
#engageug



Conclusion

- Exciting and Interesting, but a huge distraction
- Good opportunity to consolidate products (but remember MVP)
- XPages gave you much more than you realised
- Much more work that you think
- We are very proud of what we have achieved