

 Adaptive

Open-sourcing what's
inside a bank

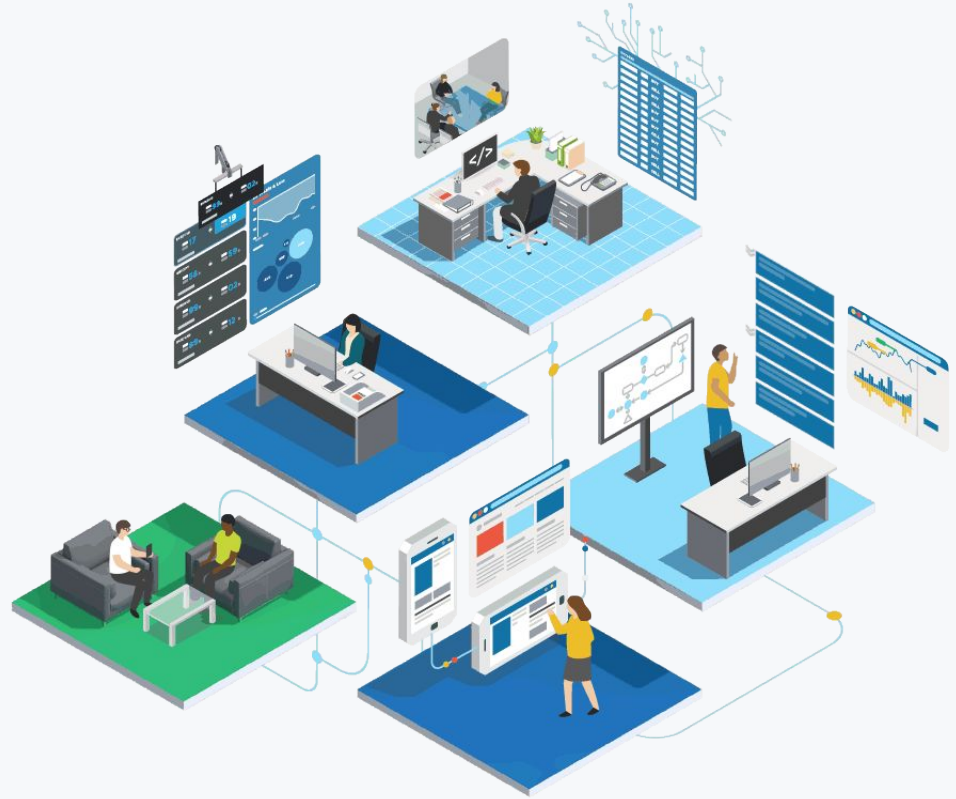
Or, what *is* competitive advantage?

What we do

The real-time trading experts.

We were founded based upon the belief that all commerce will transition to digital, and that the best experiences will be real-time

We **design, build, and operate** business led technology solutions utilising cutting edge techniques



Who we are

The real-time trading experts

Our history

- Founded in **London in 2012**
- Rapid growth since then with over **175 consultants** globally
- Offices in **London, New York, Barcelona and Montreal**
- Focussed on **financial services, capital** and **commodity markets**

We design and deliver solutions into complex organisations

- Delivered Client, Sales and Trading facing platforms
- Real-time data distribution from the cloud and on-prem
- Event-driven collaborative workflows and decision support tools
- Immersive, intuitive user interfaces for desktop and mobile

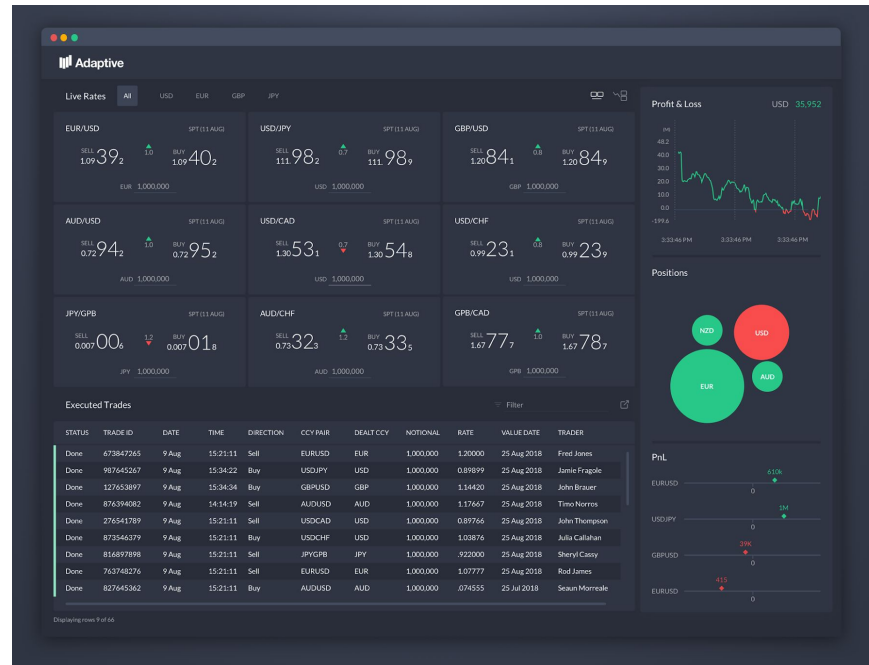


Our unique blend of IP, business insight, design, technology and change management provides focused, **experienced teams** with specific **technical expertise** and **deep business knowledge**, **delivering lasting competitive advantage** for our clients

Agenda

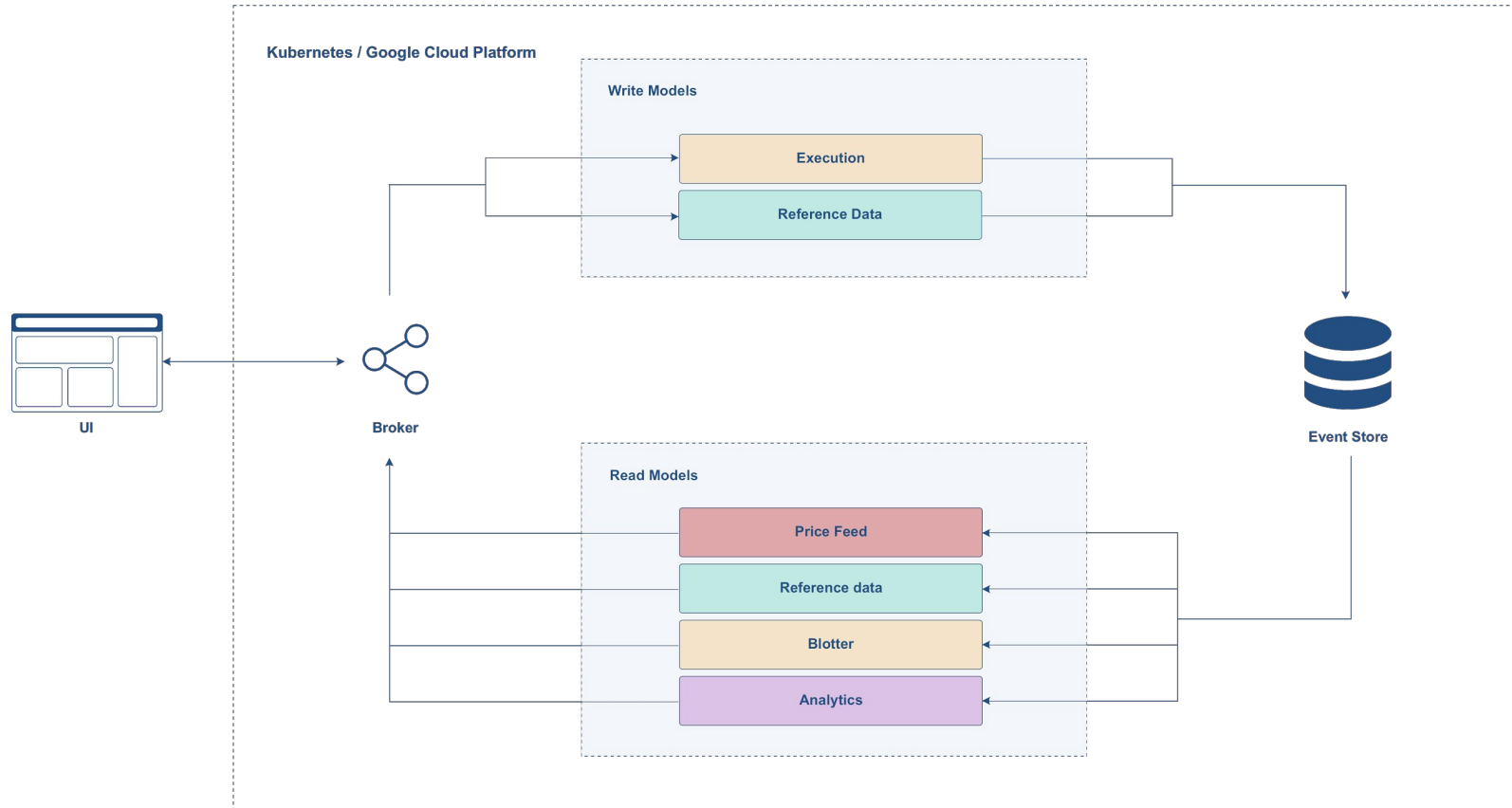
- An open source bank stack
- What is competitive advantage
- Why open source?
- What we see as being open sourced
- Why we think its wrong
 - Because we need to think differently about what to open source

A recognisable (open source) banking stack



<https://github.com/AdaptiveConsulting/ReactiveTraderCloud>
<https://web-demo.adaptivecluster.com/>

High Level Architecture



What is competitive advantage?

- Business operating model
- Process and culture
- Technical infrastructure

What is competitive advantage?

- **Business operating model**
 - Latency arbitrage, universal service, product specialisation
- **Process and culture**
 - Development model and tooling to improve developer productivity
- **Technical infrastructure**
 - UI tooling, messaging and APIs, workflow engines, high availability patterns

Why do open source

Why do open source

- **Soft reasons**
 - Copying others without understanding why
 - The optics of giving back
- **Hard reasons**
 - Externalise costs
 - Reduce hiring/training costs
 - Change the way the market operates to your benefit

Copying others without understanding why

- If you're going to open source, you need to do it well
- What is open sourced needs to be reusable
- Technical, process and operating model coupling all prevent re-use

How to tell if a FLOSS project is doomed to FAIL

Credit to [Tom 'spot' Callaway](#) ...

Size

- The source code is more than 100 MB. [+5 points of [FAIL](#)]

Source Control

- There is no documentation on how to use it for new users [+5 points of [FAIL](#)]
- You've written your own source control for this code [+30 points of [FAIL](#)]
- You don't actually use the existing source control [+50 points of [FAIL](#)]

Building [From Source](#)

- There is no documentation on how to build from source [+20 points of [FAIL](#)]
- Your source is configured editing flat text config files [+20 points of [FAIL](#)]
- Your source is configured by editing code header files manually [+30 points of [FAIL](#)]
- Your source only builds with third-party proprietary build tools [+50 points of [FAIL](#)]
- You've written your own build tool for this code [+100 points of [FAIL](#)]
- And many more....

The optics of giving back

- Top 10 open-source contributors, by originating organization,
- No finance firms here...
- What are the optics of this to prospective talent we need to compete for?

#	Project	Contributors
1	Microsoft	7700
2	Google	5500
3	Red Hat	3300
4	UC Berkeley	2700
5	Intel	2200
6	University of Washington	1800
7	Facebook	1700
8	MIT	1700
9	University of Michigan	1600
10	Stanford	1600

Externalize costs

- Take an internal capability that requires continual investment, and open source it (well)
- Provide stewardship to build a community, have it become self-maintaining, and reduce your investment
- Any good candidates in our open source bank stack?

Externalize costs

- A major library, recently open-sourced by one of our clients, uses Apache Arrow for in-memory data. A new version of Arrow was released, which changed its API
 - Unprompted, the Arrow team raised a PR to update our clients' open-sourced library, to use the new API
- Now imagine this at scale...

Reduce hiring/training costs

- An open-sourced component with wide adoption could mean:
 - New hires know the tech stack already, having worked with it elsewhere
 - Your hiring pool becomes much bigger
- Time-to-productivity will be dramatically shorter
- The candidate might already have contributed to your business in a meaningful way before they joined, and this can broaden your hiring pipeline

The graduate pipeline and university engagement

- Open-sourcing would transform the graduate pipeline
 - Students could work with real platforms, and real components
 - They could even contribute towards those components
 - And they'd be far more inclined towards applying to the firm that 'gave back'

Change the way the market operates

- Open sourcing a significant component and forming an ecosystem around it allows you to force the market to evolve.
- Evolution from 'custom built' to 'utility / commodity' through 'product'
 - Simon Wardley
- When a capability becomes utility, previously uneconomical activities become viable
 - If you can force a capability out of 'product' into 'utility', you can innovate and differentiate with this new spare capacity, or take advantage of new supply for your own benefit

What we see as being open sourced

What we see as being open sourced...

- **Proprietary solutions, often highly coupled to:**
 - Internal technical infrastructure
 - A bank's culture or procedural environment
 - A bank's operating model
- **This is not zero benefit - it gives soft benefits. However:**
 - Limited value to a wider audience
 - Low adoption
 - Won't change the game

Why we think its wrong

Why we don't think this is the right thing

- Silicon Valley firms open source a layer of business or technical architecture (or both)
 - Sometimes they help build a business around it
 - But often it adds gasoline to the adoption process, or changes market evolution
 - Externalises costs - internal infrastructure often becomes revenue generating itself

Why we don't think this is the right thing

- **Some examples of major open-source projects from Silicon Valley:**
 - **Google:** Kubernetes, React, PyTorch, Open Compute Project
 - **Facebook:** Asgard, TensorFlow
 - **LinkedIn:** Angular, Android, Titus
 - **AirBnB:** Kafka, Atlas
 - **Netflix:** Conductor, Cassandra

Why we don't think this is the right thing

- **Some examples of major open-source projects from Silicon Valley:**
 - **Google: Angular, Android, Kubernetes, TensorFlow**
 - **Facebook: Cassandra, React, Open Compute Project, PyTorch**
 - **LinkedIn: Kafka**
 - **AirBnB: AirFlow**
 - **Netflix: Asgard, Titus, Conductor, Atlas**

We need to think differently

We need to think differently

- Banks should consider top down what to make a utility, and where to compete and differentiate
- Banks should open source plumbing, and build ecosystems around it
 - Ecosystem should be encouraged to generate revenues through service and support
- Any technology open-sourced needs to be fit for capital markets world
 - Silicon Valley technology won't necessarily fit with engineering constraints
 - Bank plumbing may not have utility for other industries

We need to think differently

- The regulatory environment is pushing towards an ever-increasing standardization
 - Requirements are uniform across industry, and standardized
 - And this trend will continue
 - There are firms that can support, and who know the business well
 - It fits perfectly in the capital markets space
 - And can change the game, by competing with expensive vendors
- So... are regulatory platforms an example of open sourcing a 'clear layer of your business / technical architecture'?

We need to think differently - a roadmap to an open source utility

Closed-Source

The traditional model. A team of developers, working internally, produce a platform tightly coupled to the banks internals, then continue to support it

Mutualization

Several firms collaborate on a solution, mutualizing the costs, while moulding their processes internally to fit the newly defined model. Works extremely well with regulatory platforms, which are pre-standardized

Commodity / Utility

A firm, or a consortium of firms, produce a platform which is then sold into other firms. Parts of the tech stack can be open sourced, including standards and plumbing

Open Source

Fully open-sourced, potentially previous versions (Google does this well).

Encourage ecosystem that generates revenue - support and customisation provided by new or third party firm.

Thanks

Follow us on:



@weareadaptive



info@weareadaptive.com



github.com/AdaptiveConsulting

