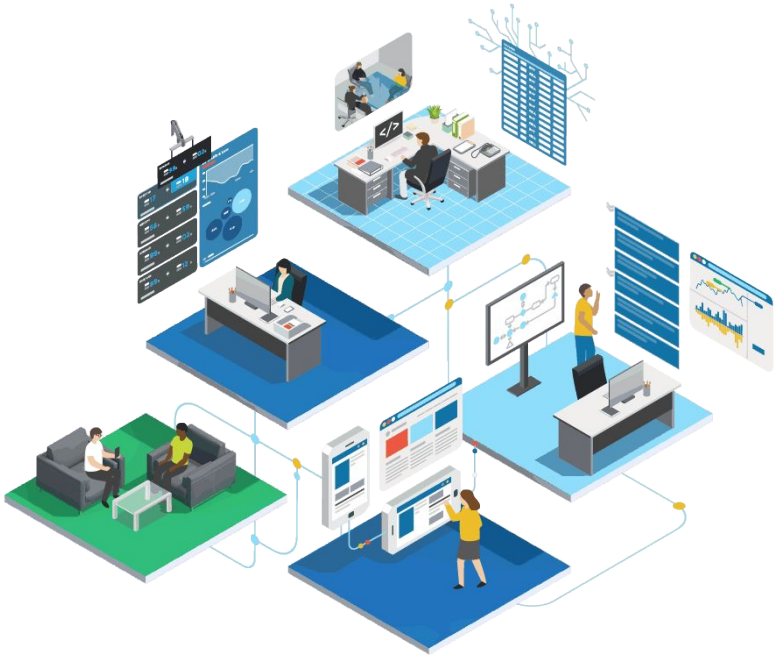


Adaptive

Real-Time Streaming UIs

Matt Barrett
Co-founder @ Adaptive

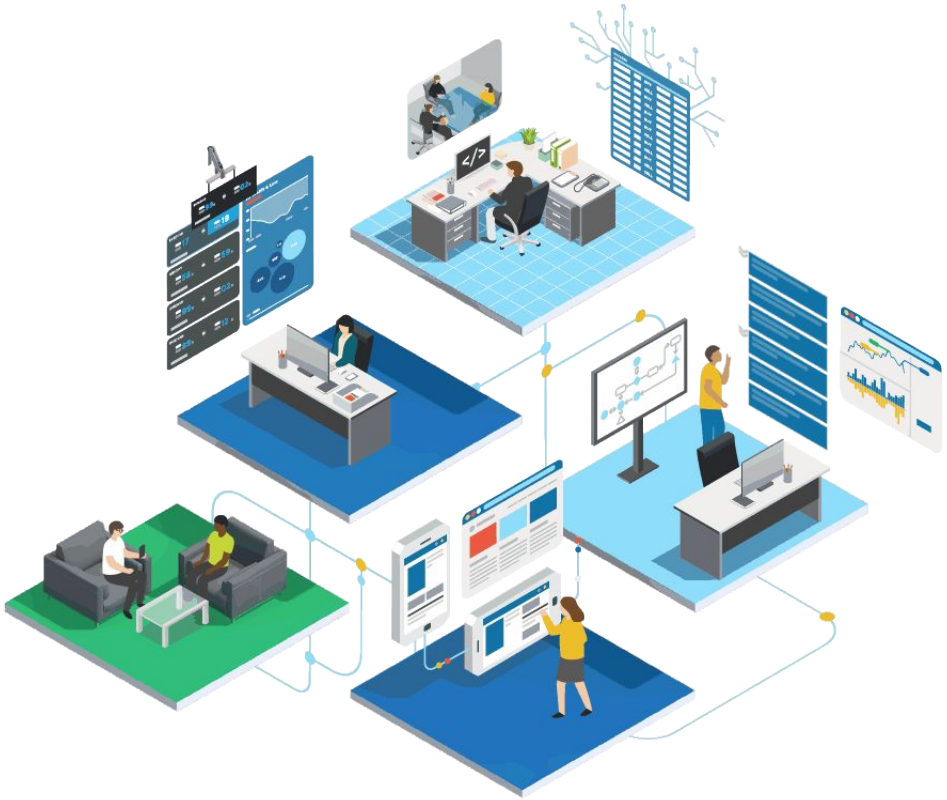
Bhavesh Desai
Head of UI @ Adaptive

NDC { Oslo }

Who are we?

The real-time trading experts.

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



Adaptive

Market forces and *real-time* trading *technology* are fundamentally changing the way business is conducted within financial services, capital and commodity markets.

Adaptive

Market forces and *real-time* trading *technology* are fundamentally changing the way business is conducted ~~within financial services, capital and commodity markets~~ *everywhere*!

Adaptive

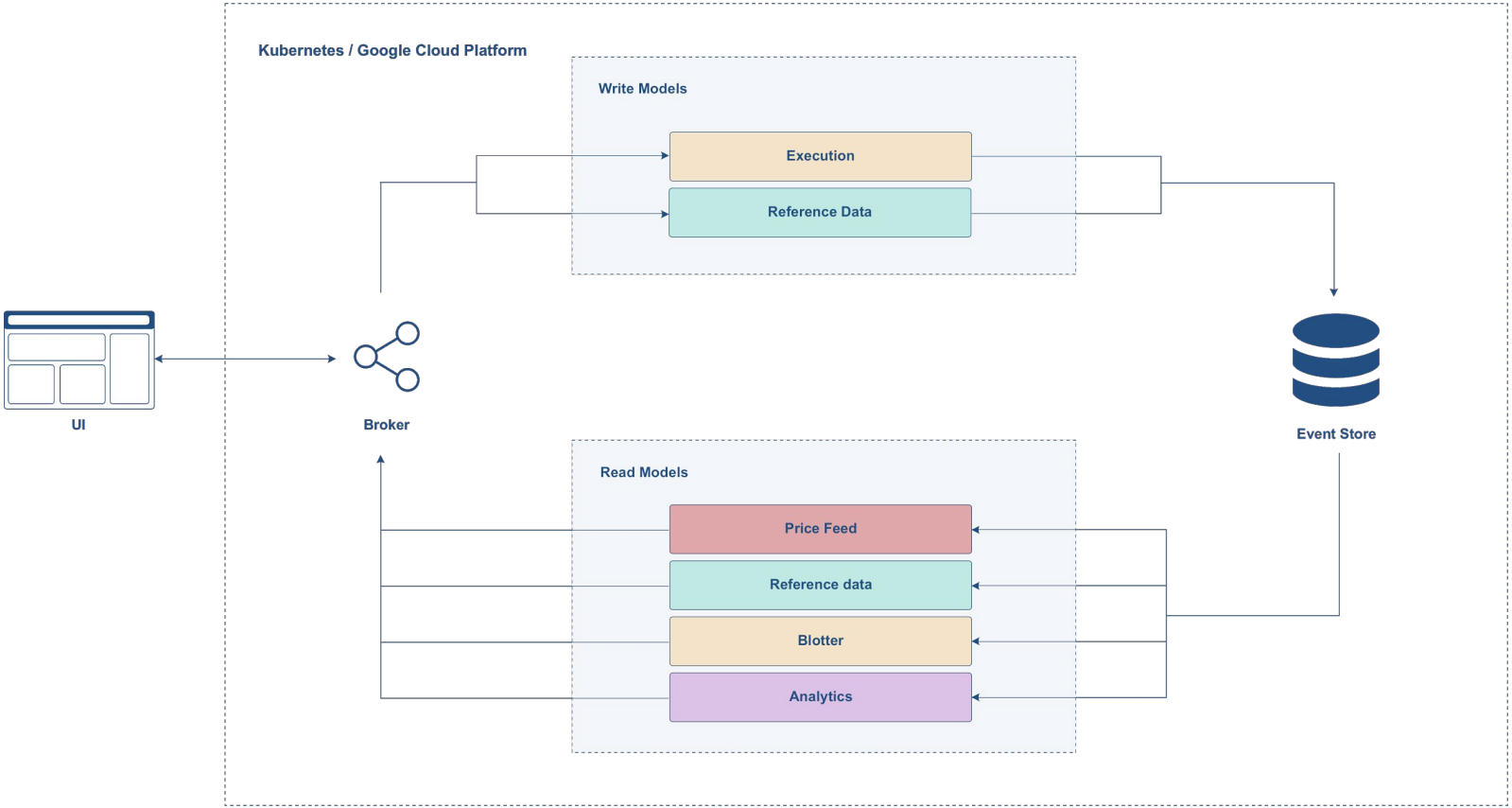
Streams and UIs

- Real-time streaming web applications
- Embracing streams and managing complexity
- Streams, time and the business domain
- Performance
- Questions

Demo

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

High Level Architecture

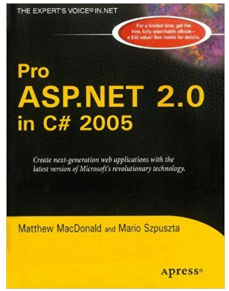


Adaptive

Real-time streaming web apps

Adaptive

The Early, Difficult Days



Ext JS

jQuery
write less, do more.

Java Swing

WPF
Microsoft® .NET
Windows Presentation Foundation

JavaFx

JSP

ASP.NET MVC

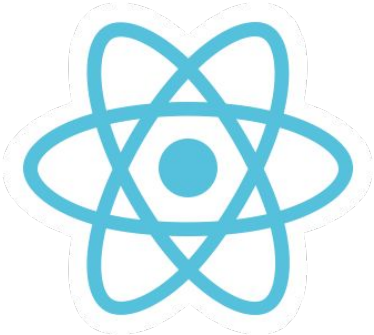
The RIA Era

Today’s UI Frameworks

Angular 2 +

20%

React

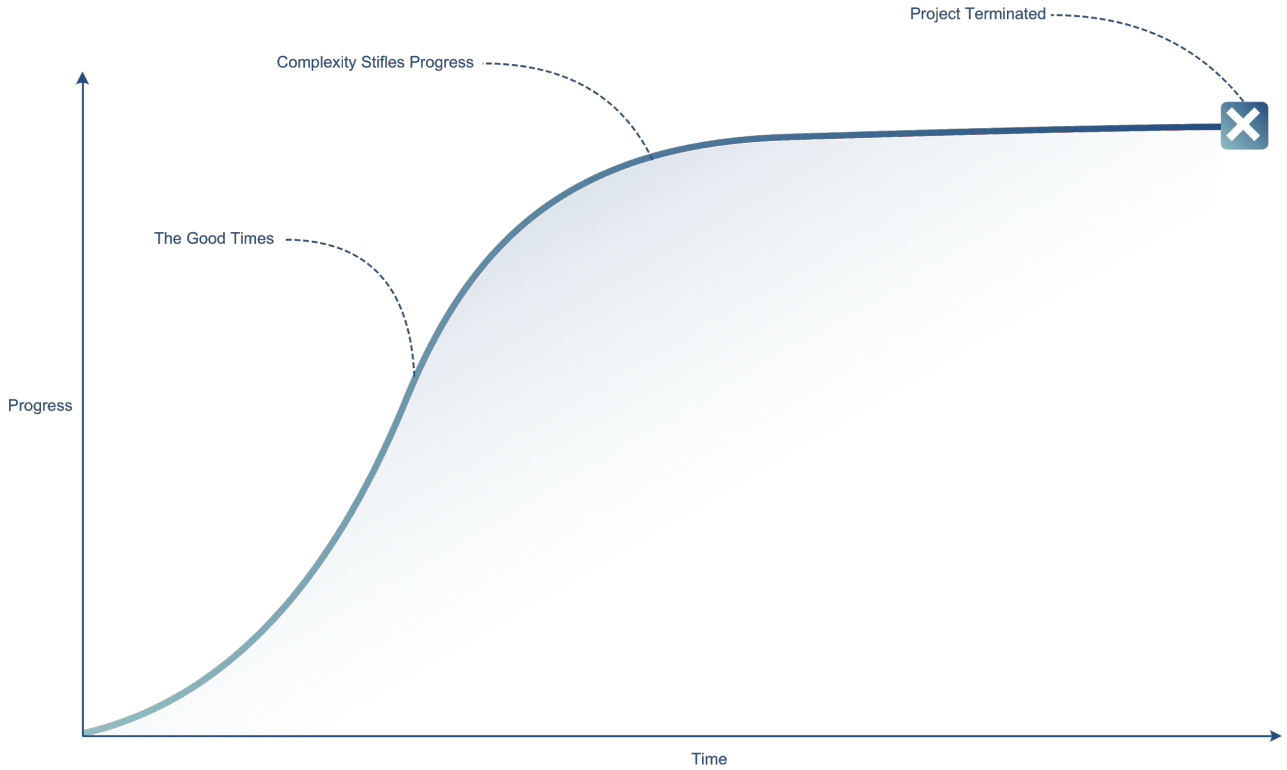


80%

Complexity In Trading Applications

- Lots of different things traded..
  - FX, Equities, Fixed Income, Physical Commodities (Vanilla pods!), Energy, etc.
- Packaged in different ways..
  - Spot, Forwards, Swaps, Options..
- Traded in different ways..
  - Request for Quote, Indication of Interest, Order, Executable Streaming Price
- Large amounts of information on screen
- Streaming data
- Time sensitive
- Performance issues have large, possibly existential business impact

The Cliff of Complexity

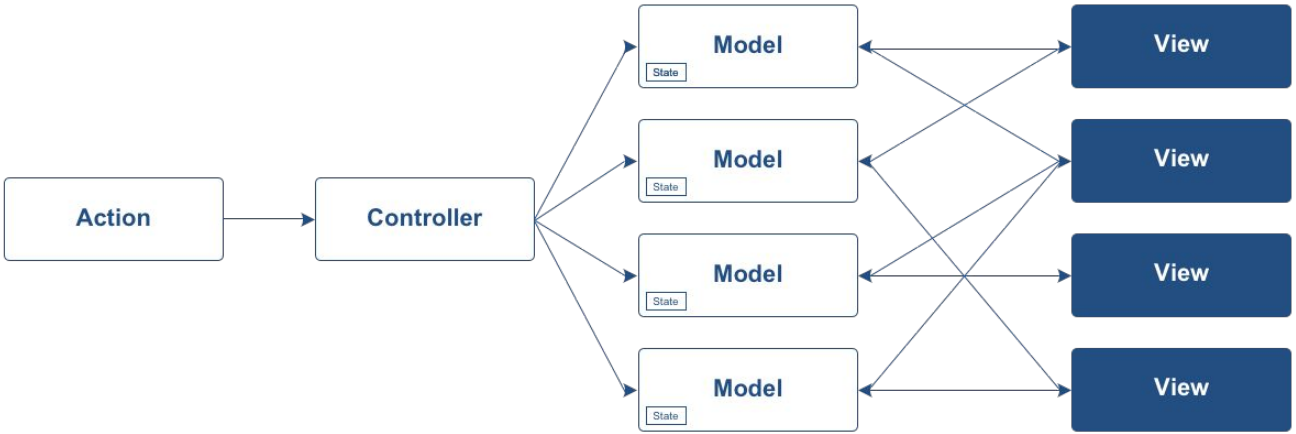


Adaptive

Embracing streams and managing complexity

Avoid Client Side MVC

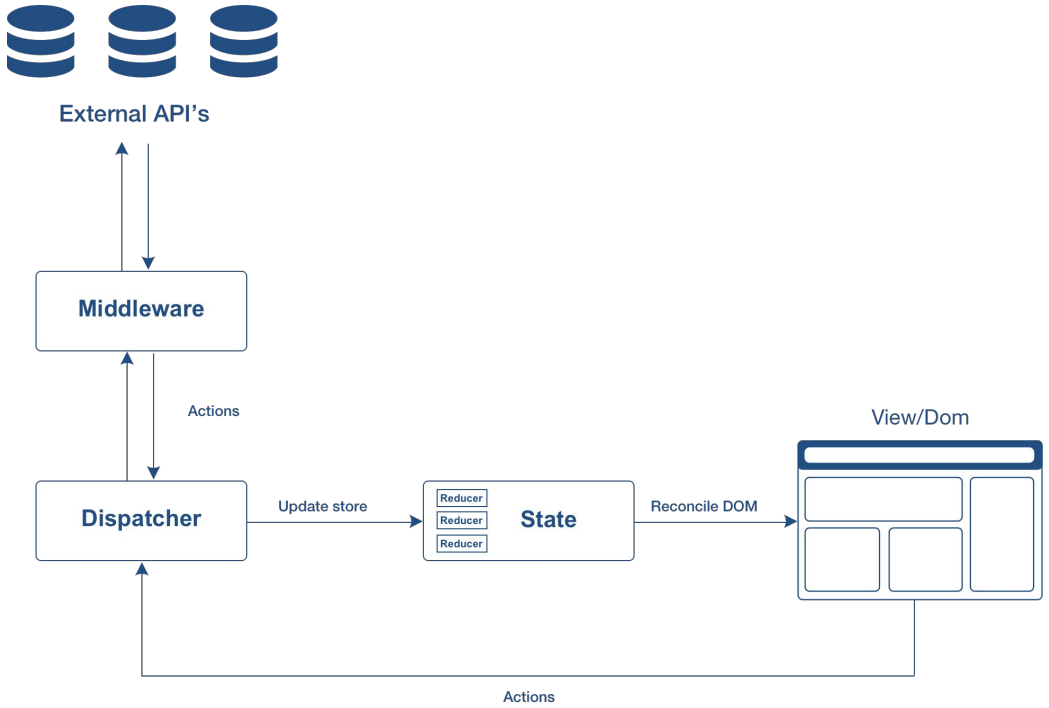
- State spread across multiple locations. Hard to keep in sync
- Two-way data flow can be confusing
- Hard to track the cause and effect of events



Redux

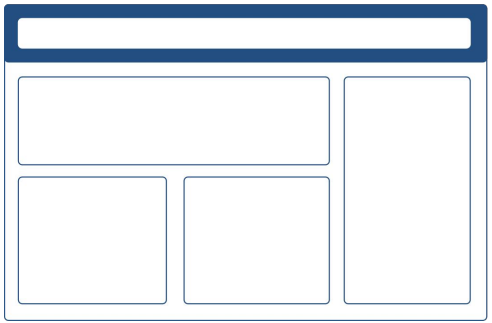
Event-sourcing on the client

- One place to store state
- Data flows in one direction
- Widely used
- **Simple programming model**



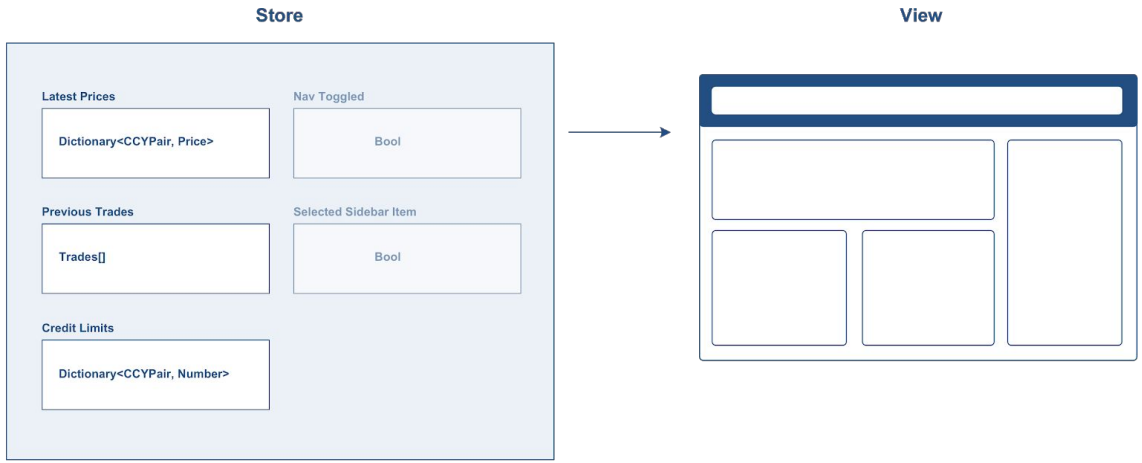
Start with a Template

View

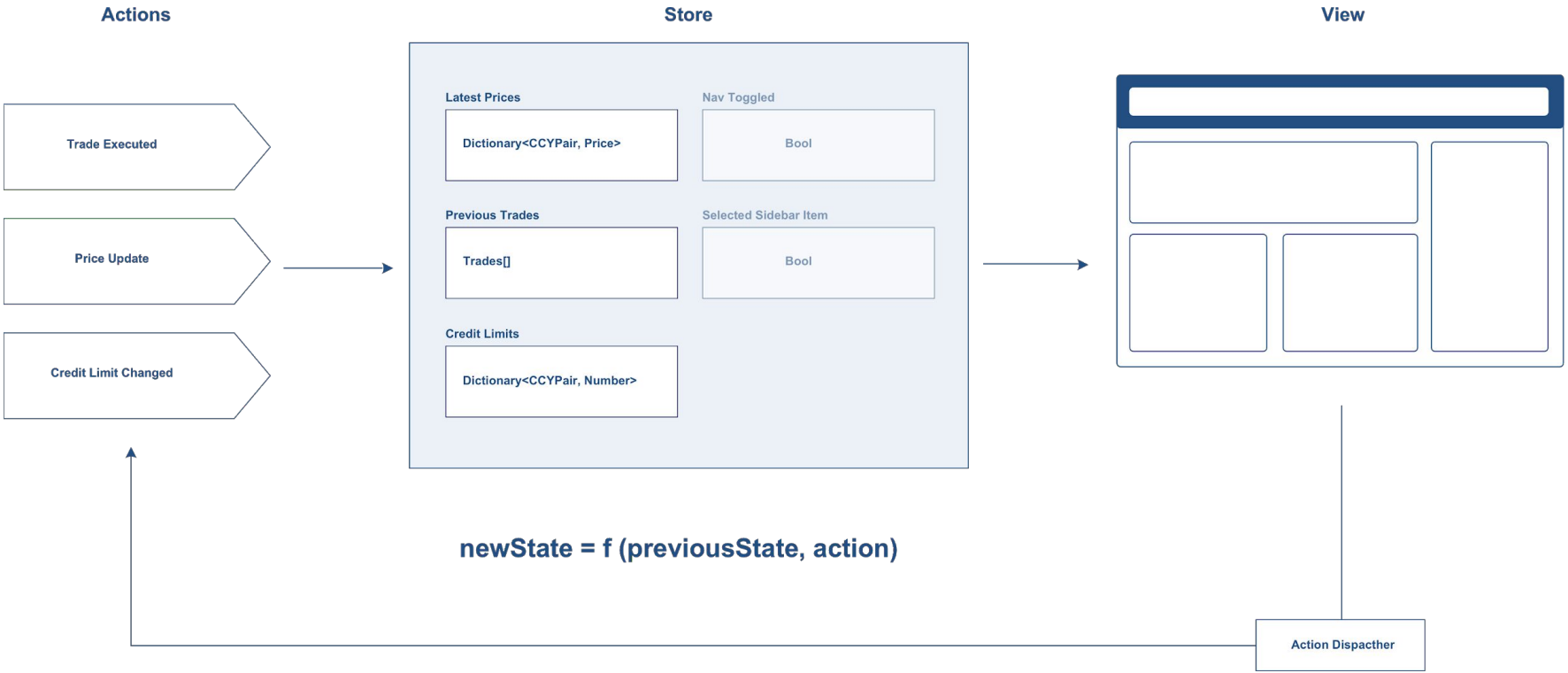


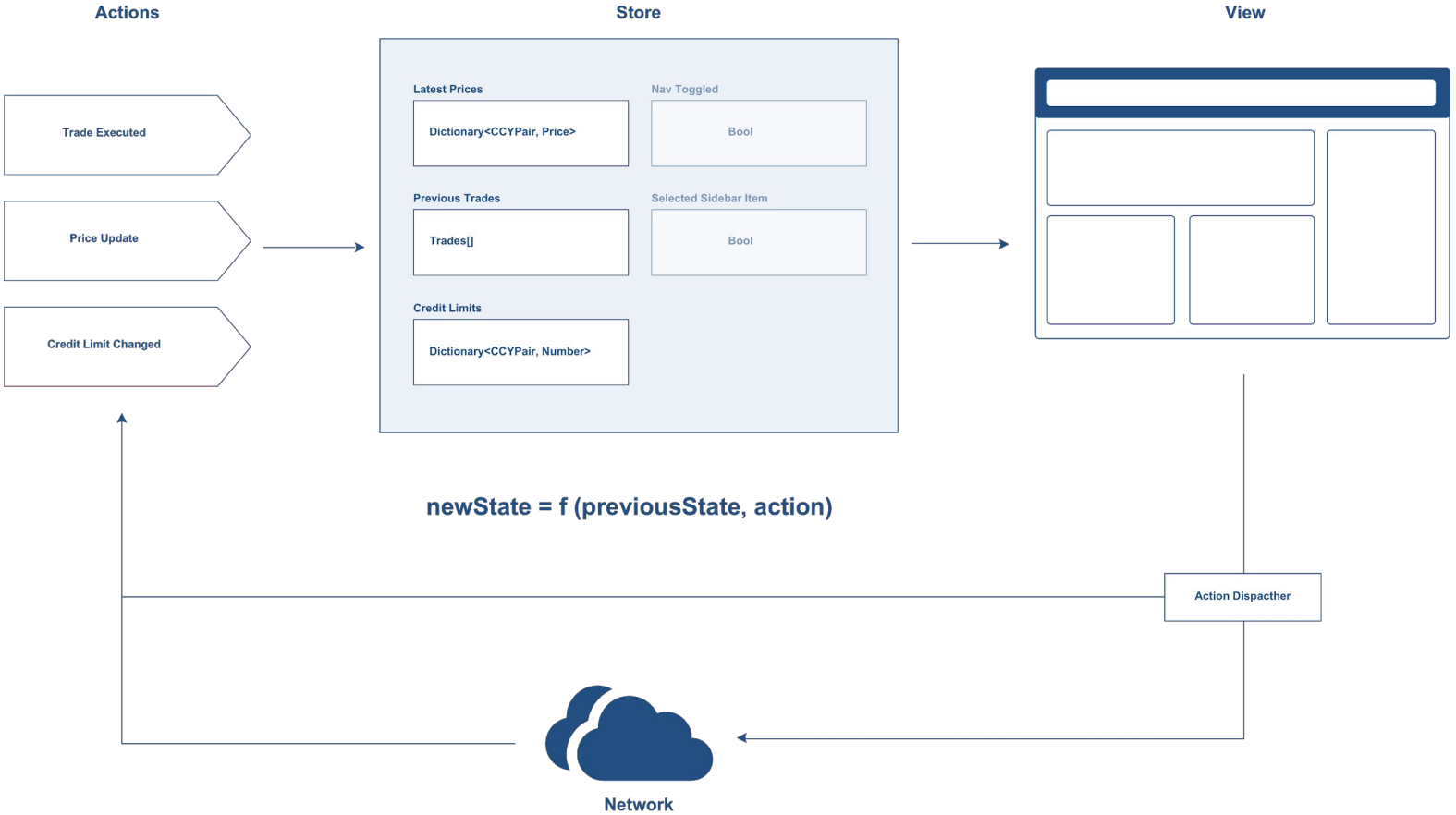
Adaptive

Model the Applications State



Model Actions that can change our state




Actions
Store
View
Trade Executed
Price Update
Credit Limit Changed
Latest Prices
Dictionary<CCYPair, Price>
Nav Toggled
Bool
Previous Trades
Trades[]
Selected Sidebar Item
Bool
Credit Limits
Dictionary<CCYPair, Number>
newState = f (previousState, action)
Action Dispacther
Network

Adaptive

Middleware

Process of calling into the outside world

Action -> Action(s)

- Where we handle asynchronicity (Network requests or Timers)
- Redux-observable (Netflix) allows us to write middleware using RxJS

Middleware

Process of calling into the outside world

Action -> Action(s)

- Where we handle asynchronicity (Network requests or Timers)
- Redux-observable (Netflix) allows us to write middleware using RxJS

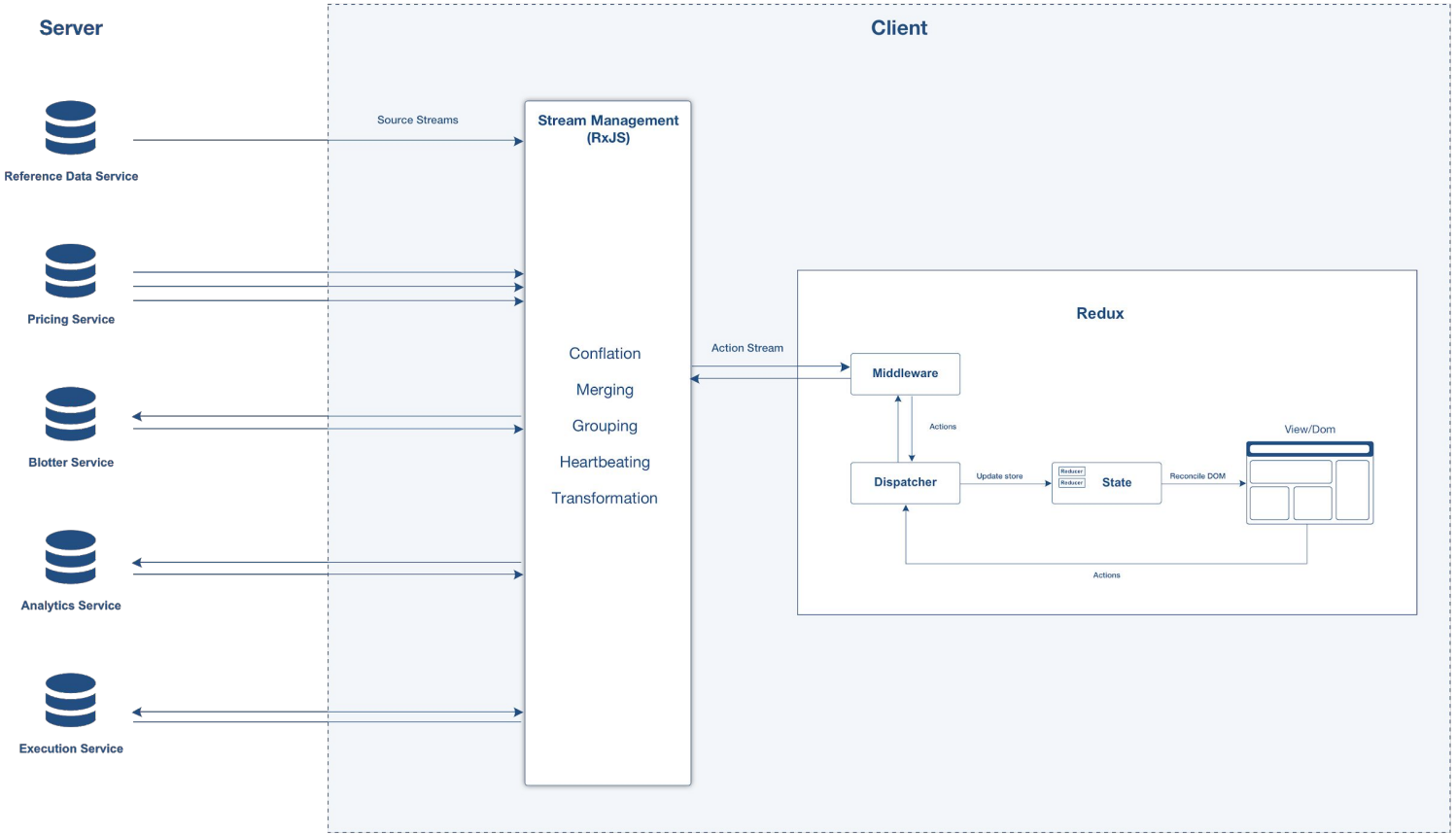
RxJS

Reactive Extensions Library for Javascript

- Library for managing and manipulating streams
- Easily compose asynchronous or callback code
- Well suited to the financial domain as most services are streams
- All through Reactive Trader Cloud

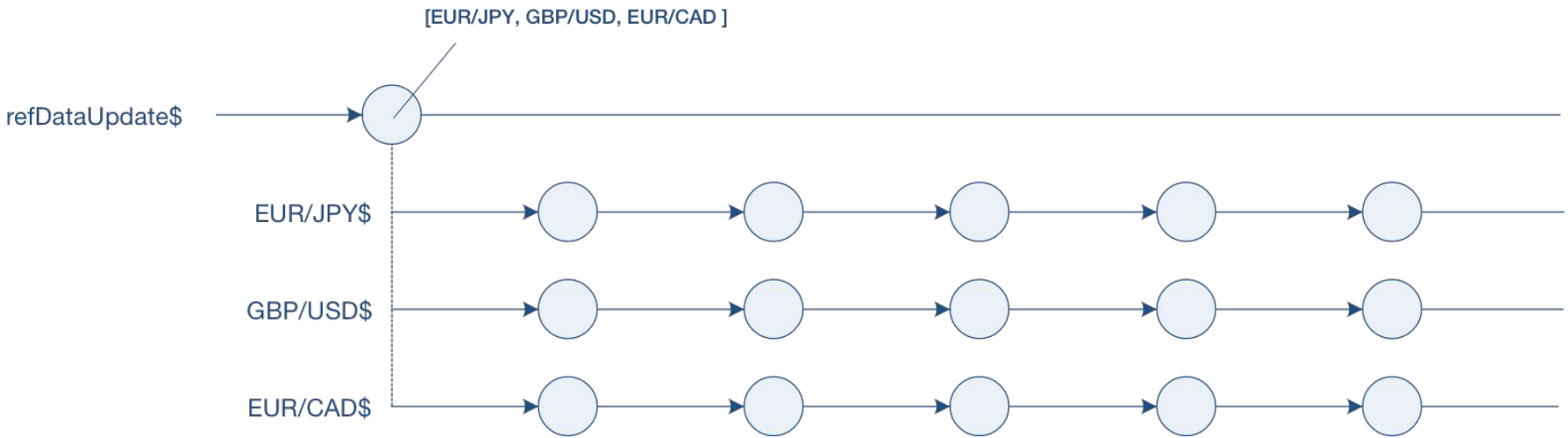
Streams, time and the business domain

Reactive Trader Architecture



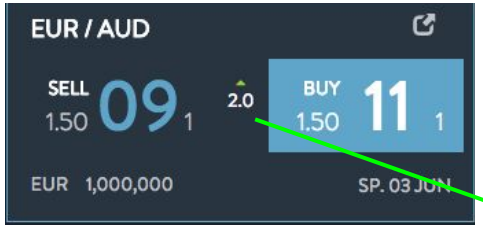
Adaptive

Price Streams



Processing Price Streams

Price movement indicator

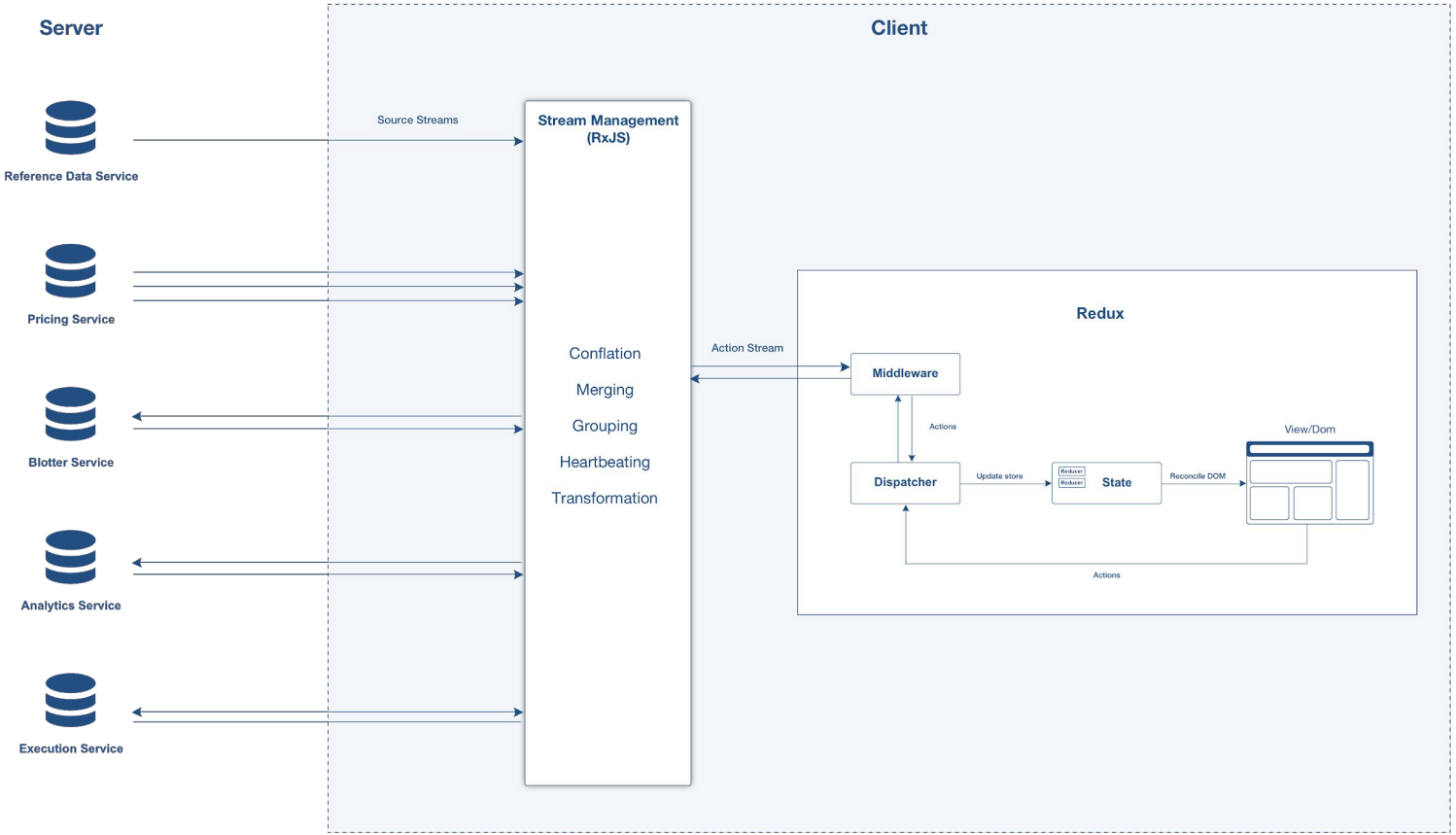


Stale price


```
private static createSpotPriceStream = (serviceWithMinLoad: ServiceClient, request: Request) => {
  log.debug( message: `Subscribing to spot price stream for [${request.symbol}]`)
  return serviceWithMinLoad
    .createStreamOperation<RawPrice, Request>(ServiceConst.PricingServiceKey, getPriceUpdatesOperationName, request)
    .pipe(
      retryWhen(
        retryConstantly({
          interval: 2000
        })
      ),
      map(adaptDTO),
      scan<SpotPriceTick>( accumulator: (acc, next) => ({
        ...next,
        priceMovementType: PricingService.getPriceMovementType(acc, next)
      })),
      debounceWithSelector<SpotPriceTick>(MS_FOR_LAST_PRICE_TO_BECOME_STALE,  itemSelector: item => ({
        ...item,
        priceStale: true
      })),
      share()
    )
}
```

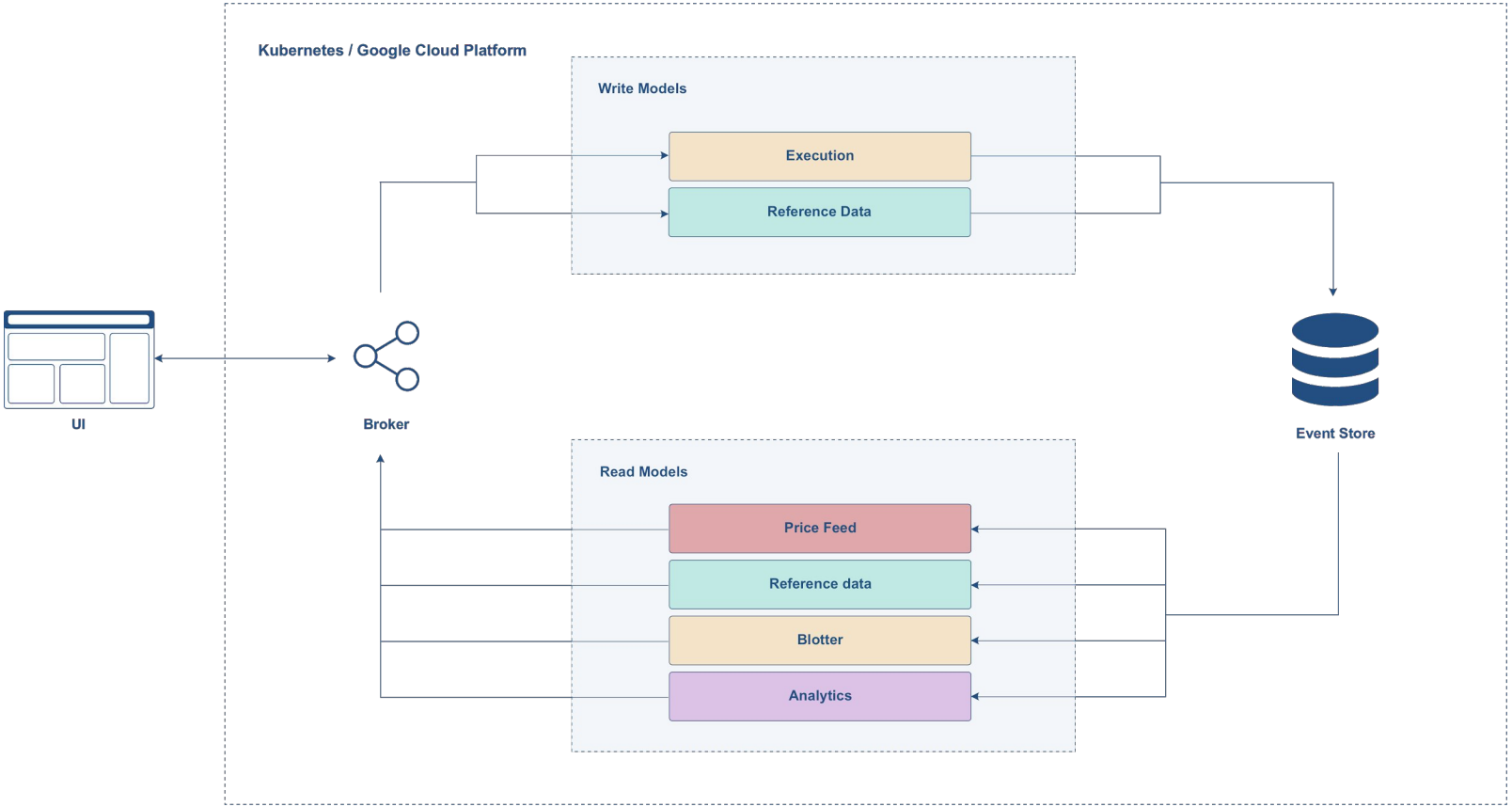
Adaptive

Reactive Trader Architecture



Adaptive

High Level Architecture



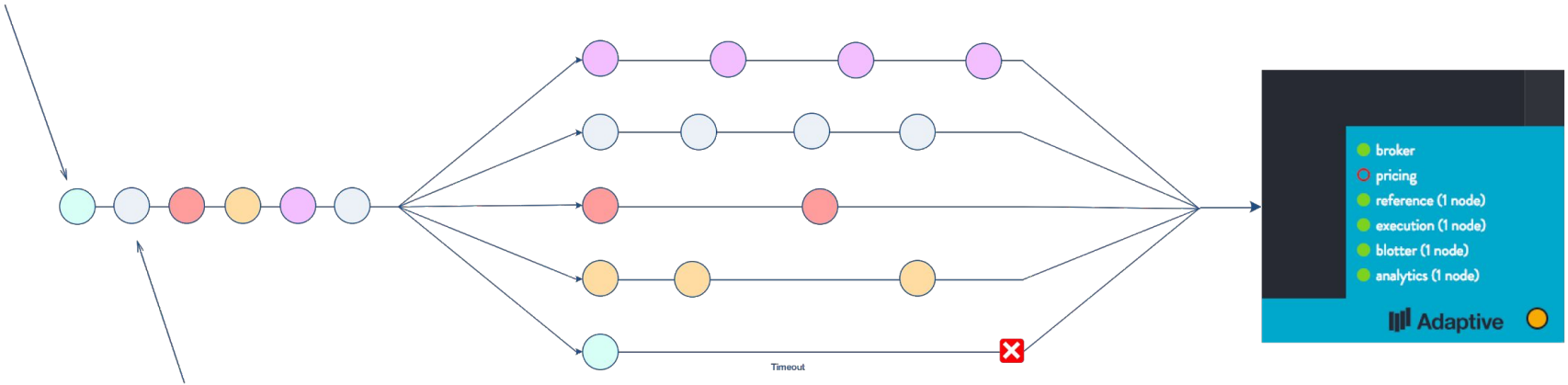
Adaptive

Demo

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

Detecting the Status of Services

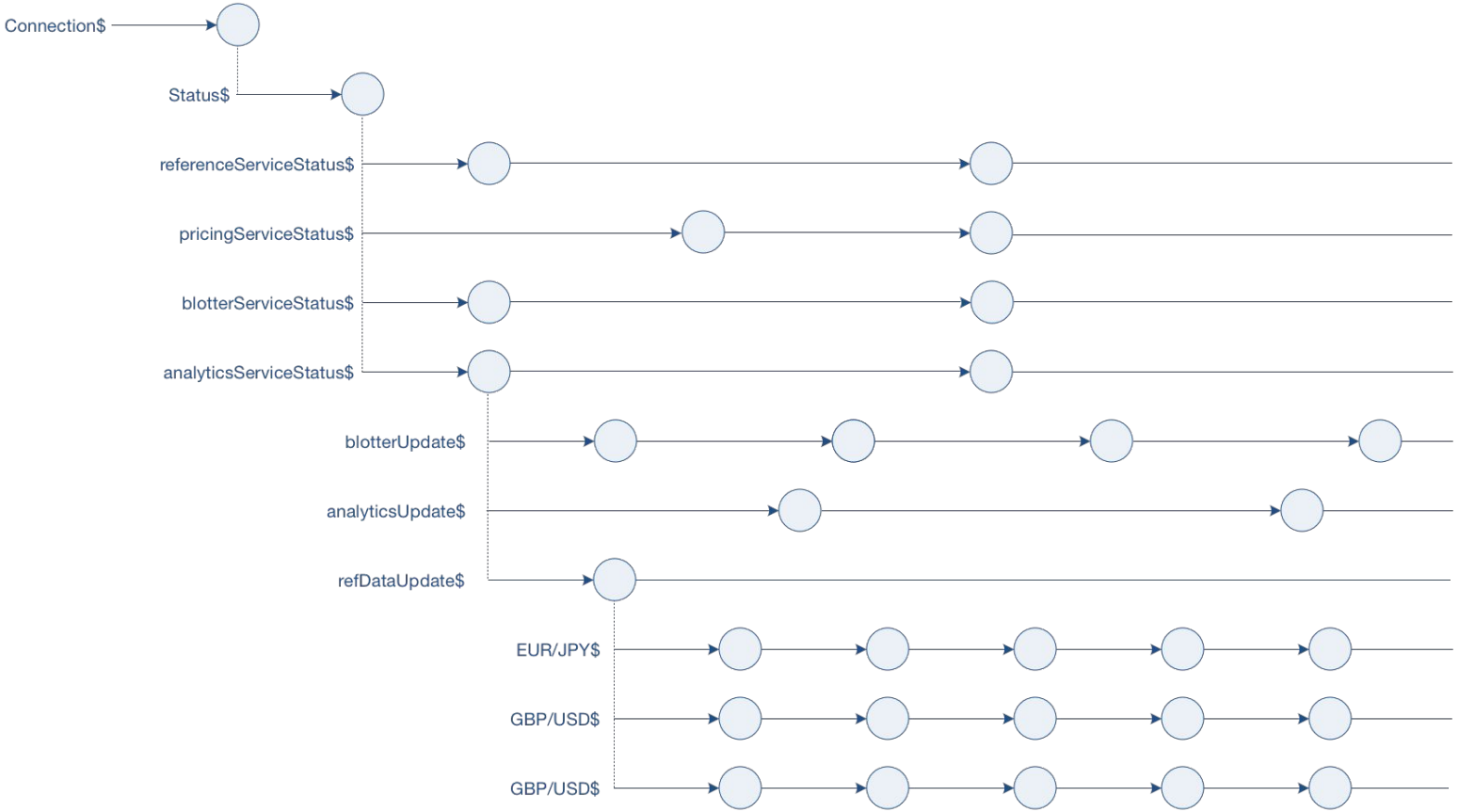
```
{
  serviceType: "analytics",
  instance: "instance01",
  load: 1.2,
  timestamp : "2018-06-07T00:00:00Z"
}
```



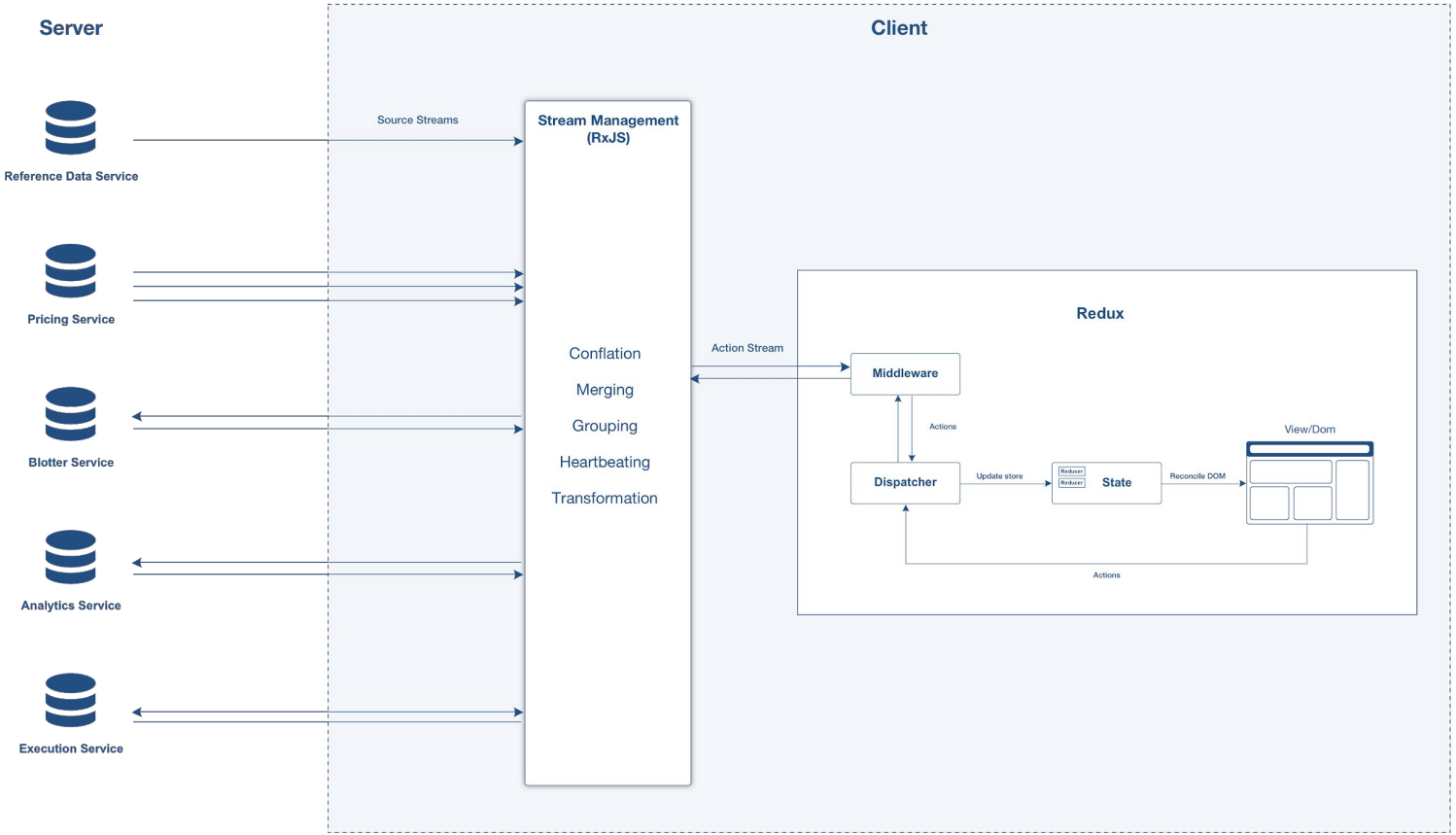

```
{
  serviceType: "blotter",
  instance: "instance02",
  load: 2.8,
  timestamp : "2018-06-09T00:00:00Z"
}
```

Adaptive

Streams of Streams



Reactive Trader Architecture



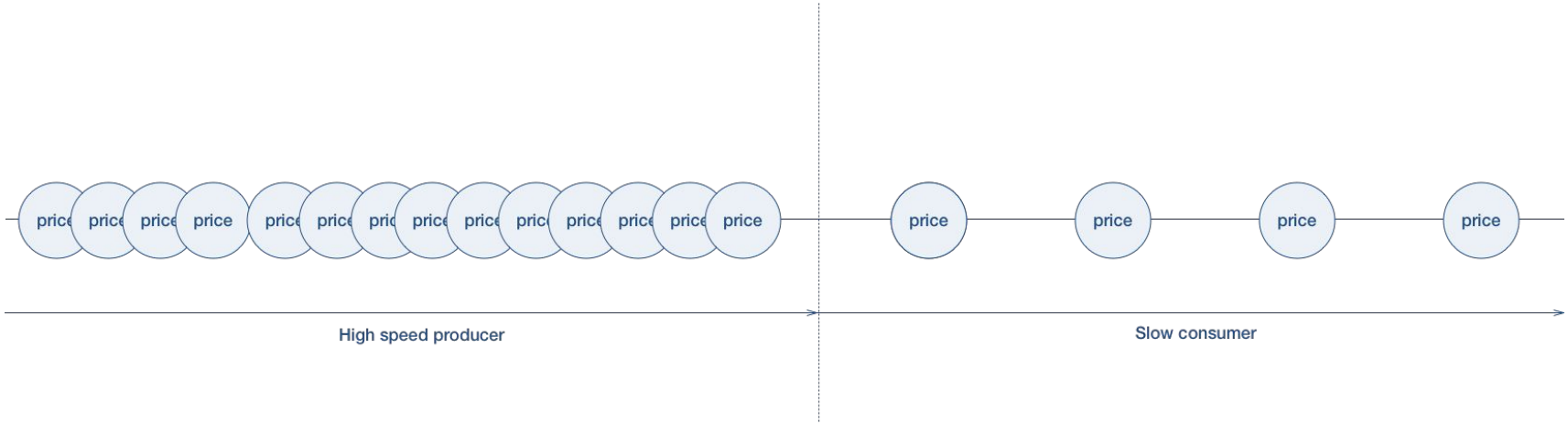
Adaptive

Backpressure

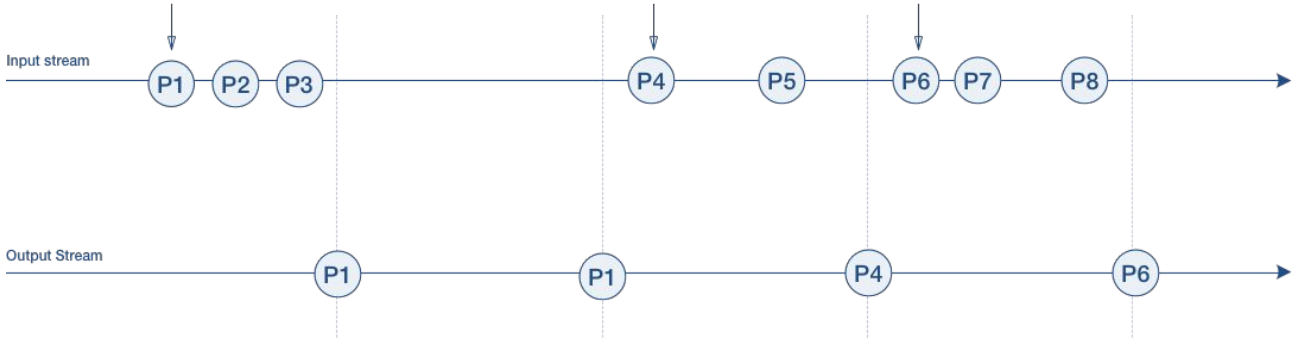
Backpressure

Consumer can not keep up with a producer due to high speed data transmission

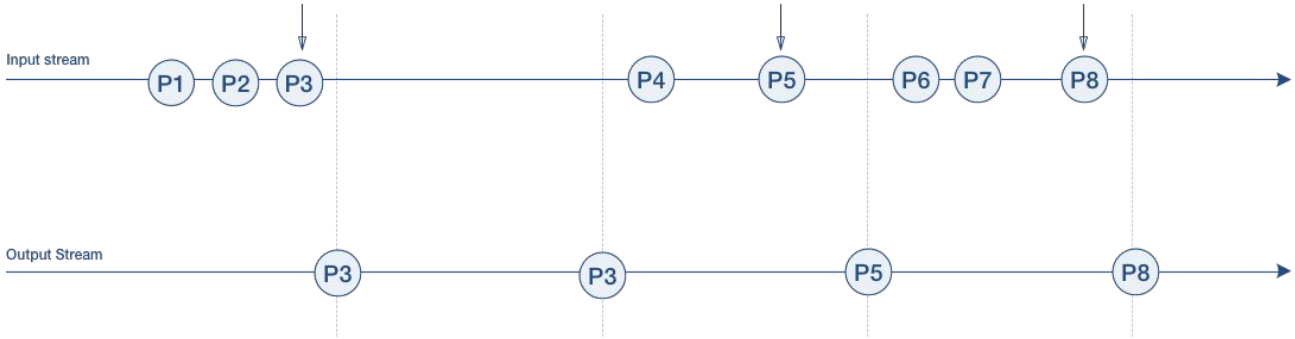
Lossy (Immutable Updates, Prices) or **Loss-less** (Mutable Updates-Deltas, Trade Blotters)

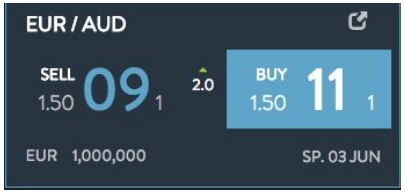


Throttling



Sampling



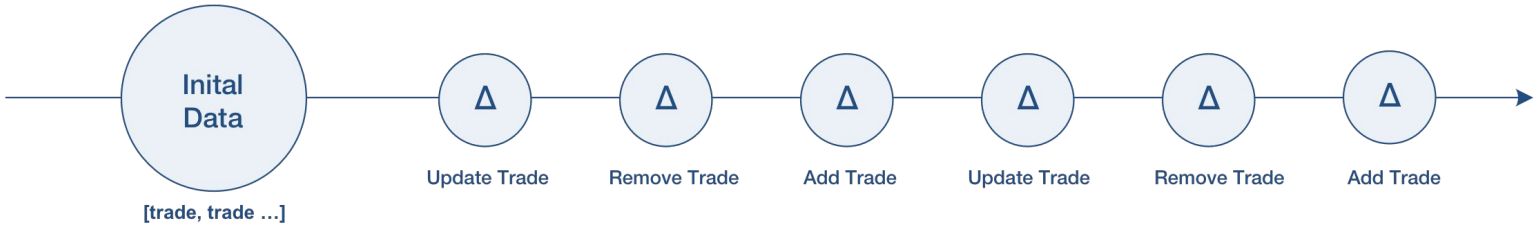


Adaptive

Loss-less streams

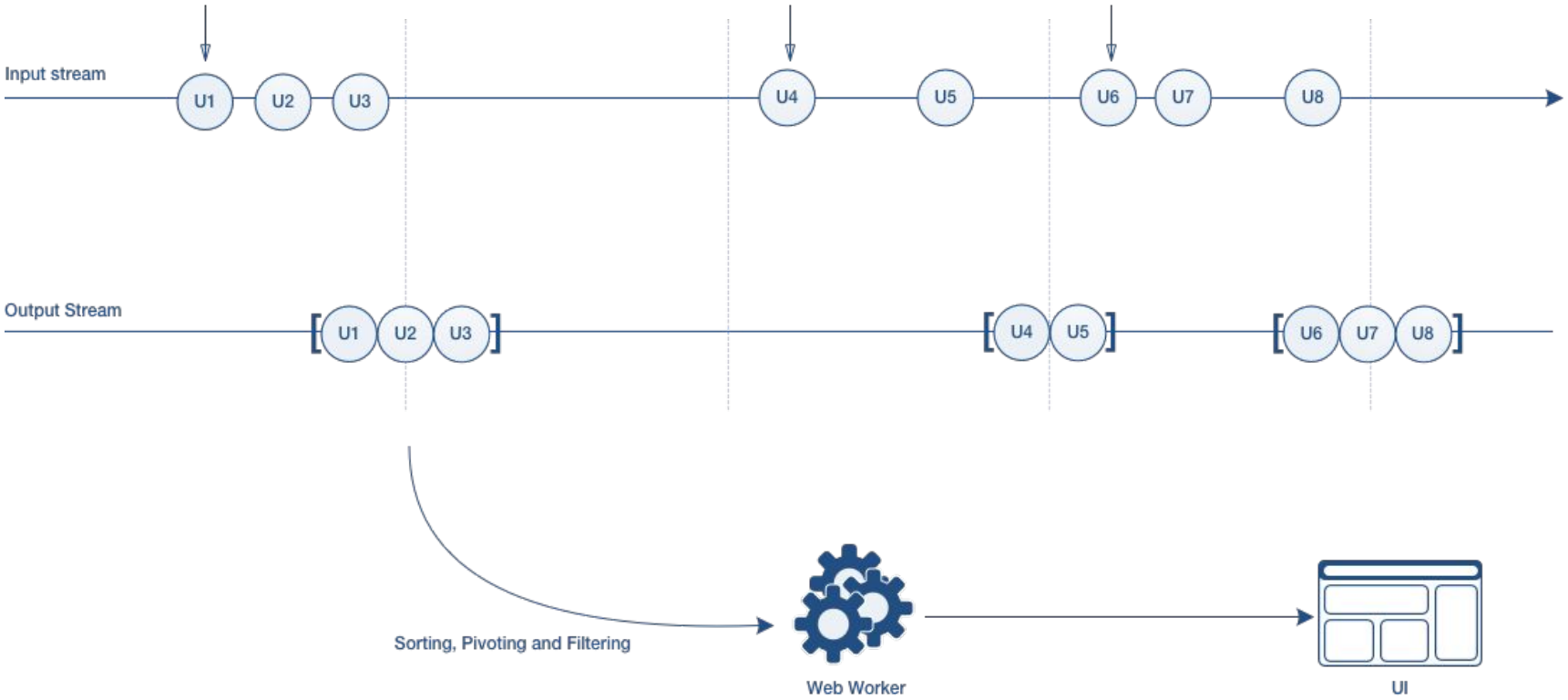
4995	Done	01-Jun 14:36:56	Buy	USDJPY	USD	10,000,000 $	121.475	04-Jun-2018	CZA
4994	Done	01-Jun 14:36:35	Buy	USDJPY	USD	1,000,000 $	121.438	04-Jun-2018	CZA
4993	Rejected	01-Jun 14:36:32	Buy	GBPJPY	GBP	1,000,000 £	184.542	04-Jun-2018	CZA
4992	Rejected	01-Jun 14:36:23	Sell	GBPJPY	GBP	1,000,000 £	184.531	04-Jun-2018	CZA
4991	Done	01-Jun 13:35:03	Buy	USDJPY	USD	1,000,000 $	121.621	04-Jun-2018	LMO
4990	Rejected	01-Jun 13:34:55	Sell	GBPJPY	GBP	1,000,000 £	184.55	04-Jun-2018	LMO
4989	Done	01-Jun 13:34:51	Buy	GBPUSD	GBP	1,000,000 £	1.51648	04-Jun-2018	LMO
4988	Done	01-Jun 13:34:50	Buy	EURUSD	EUR	1,000,000 €	1.09324	04-Jun-2018	LMO
4987	Done	01-Jun 13:34:21	Buy	EURUSD	EUR	1,000,000 €	1.09318	04-Jun-2018	LMO
4986	Done	01-Jun 13:29:33	Buy	EURCAD	EUR	1,000,000 €	1.4848	04-Jun-2018	KLA
4985	Done	01-Jun 13:29:25	Buy	AUDUSD	AUD	1,000,000 $	0.72957	04-Jun-2018	KLA
4984	Rejected	01-Jun 12:38:23	Sell	GBPJPY	GBP	1,000,000 £	184.586	04-Jun-2018	KLA
4983	Done	01-Jun 12:38:11	Sell	EURUSD	EUR	1,000,000 €	1.09544	04-Jun-2018	KLA
4982	Done	01-Jun 10:22:14	Sell	NZDUSD	NZD	1,000,000 $	0.6721	04-Jun-2018	NGA
4981	Done	01-Jun 10:07:47	Sell	NZDUSD	NZD	55 $	0.67372	04-Jun-2018	FAP

Displaying rows 50 of 50



Adaptive

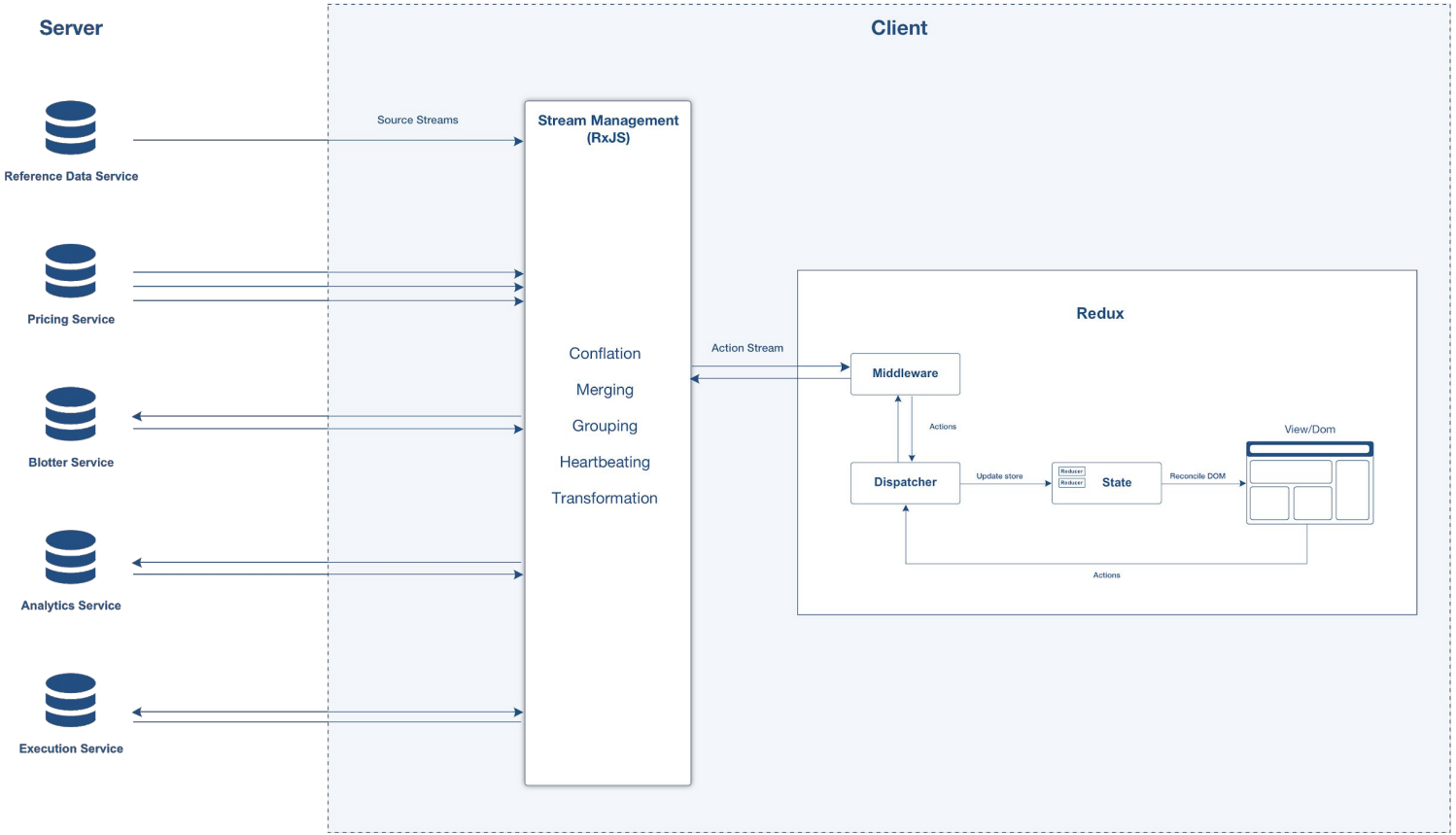
Buffering


Input stream
U1
U2
U3
U4
U5
U6
U7
U8
Output Stream
U1
U2
U3
U4
U5
U6
U7
U8
Sorting, Pivoting and Filtering
Web Worker
UI

RxJS

- Use Typescript.
- RxJS has a **high learning curve.**
- Do not use Rx if your middleware is simple.
- The Rx paradigm can be used for front-end and back-end developers to communicate
- Use Use RxJS TSLint plugin to avoid common pitfalls
- Try not to mix Pull and Push models. “Everything is a stream”
- Rx is useless if no one can read your code.

Reactive Trader Architecture



Adaptive

Performance

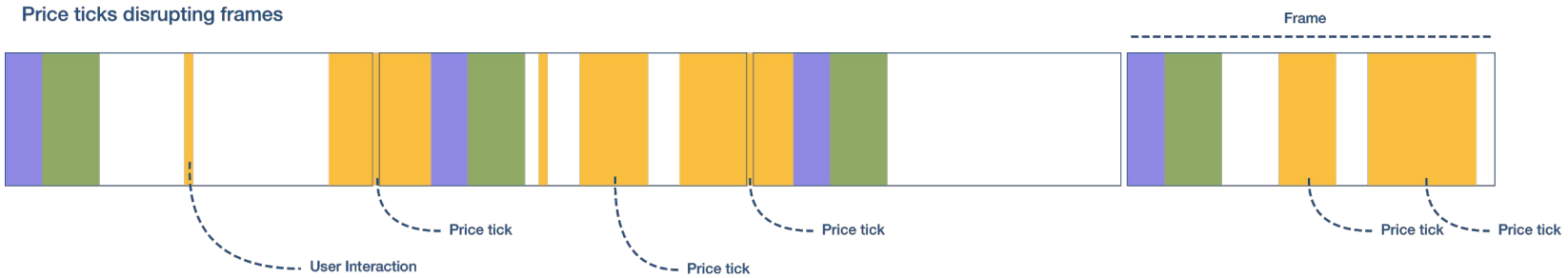
Types of Performance

- Critical Rendering Path (Async/Defer tags, Lazy loading, minification)
- JS execution (Object pooling, Data processing, micro-optimization, algorithms and data structures)
- Network Performance
- **Rendering performance**
- **Memory management**

Rendering Performance

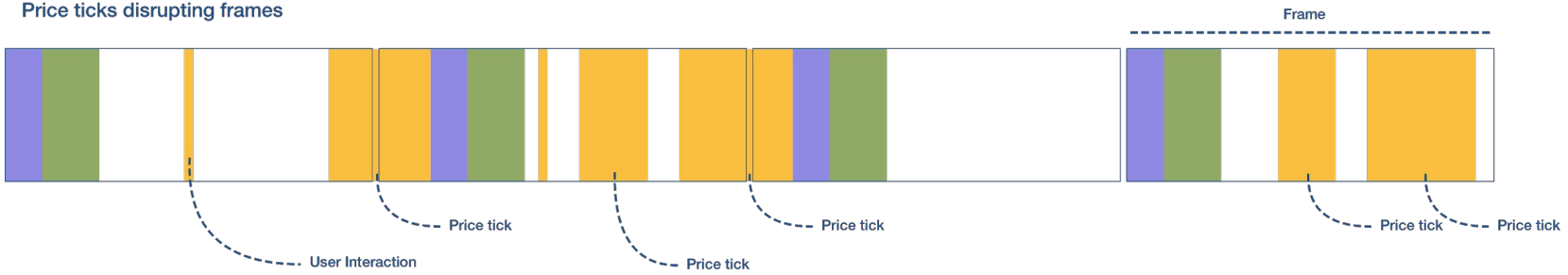
Frames with User Interactions

Rendering Performance



Rendering Performance

Price ticks disrupting frames



Using request animation frame

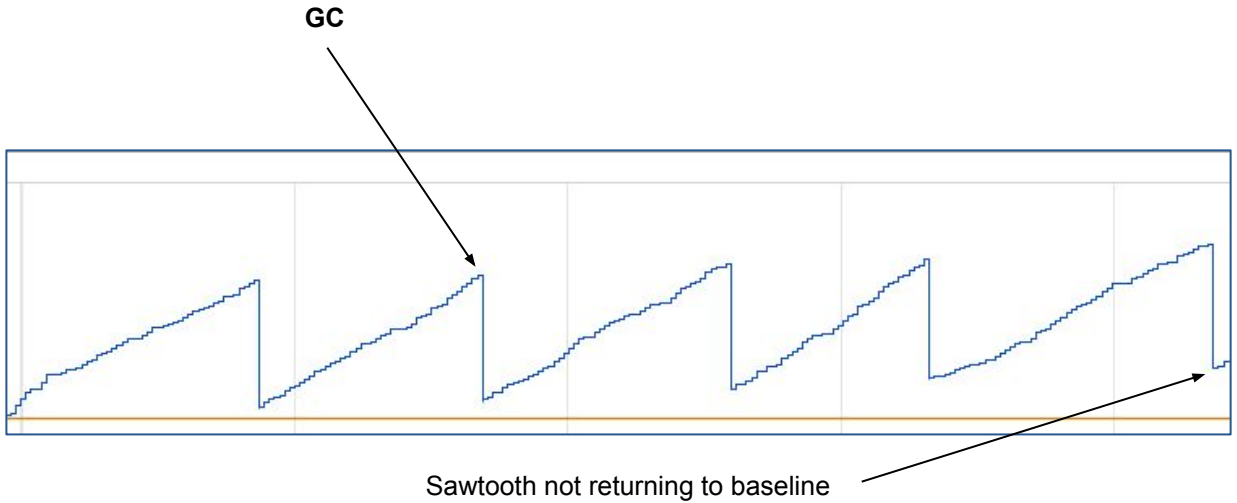
Scheduling work with Rx

Rx Scheduler	When will the work occur	Plain JS equivalent	Can block the event loop
queue	Synchronously	Synchronous code	yes
asap	Fast as possible async (when the current stack clears)	Promise.resolve().then()	yes
async	When the event loop is free	setTimeout(.., 0)	no
animationFrame	Before rendering the next frame	requestAnimationFrame()	no

Gerard Sans — Bending time with Schedulers and RxJS 5: https://www.youtube.com/watch?v=AL8dG1tuH40

Memory Leaks

- Unbounded Buffers in Rx
- Not Unsubscribing in Rx
- Detached DOM nodes
- Lingering event handlers



Market forces and *real-time* trading *technology* are fundamentally changing the way business is conducted ~~within financial services, capital and commodity markets~~ everywhere!

Adaptive

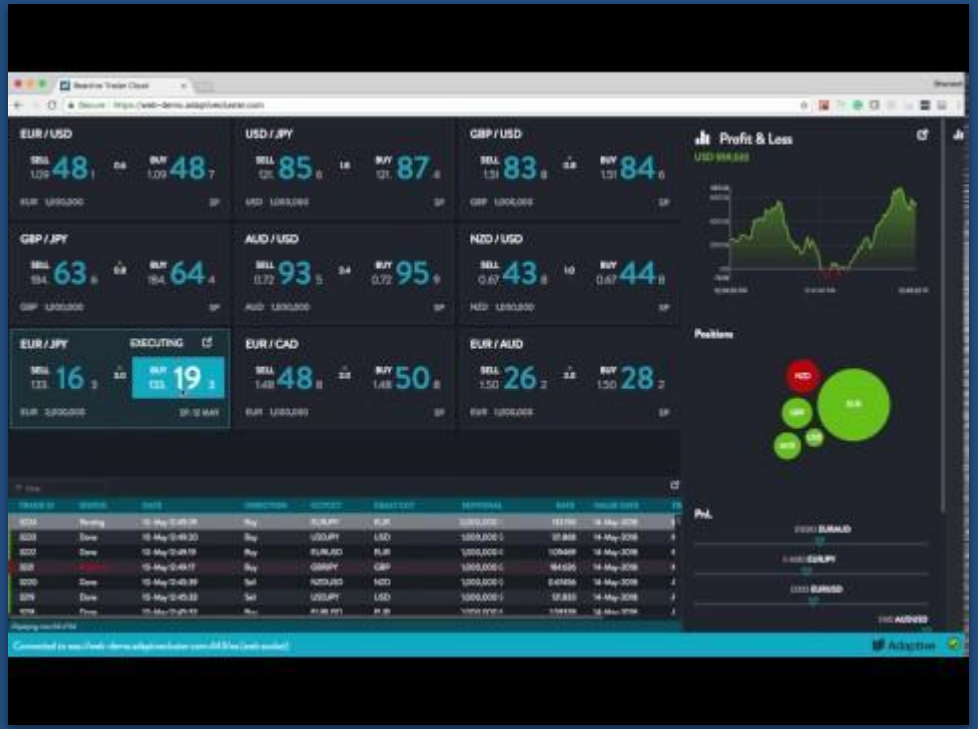
Real-time technology **enables you to change the way you build applications, to enable real time experiences for your users that are hugely compelling, and frankly, just..**

..cooler.

Thanks

Follow us on:

- @weareadaptive
- info@weareadaptive.com
- github.com/AdaptiveConsulting



Adaptive