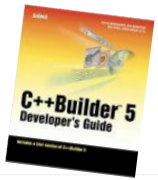
Social Media, Real Time AI and the search for Alpha

Dr Jamie Allsop — 2017

DSP background with a PhD in **adaptive framework design**



BSI | C++ Panel

pypi/cuppa

Market Microstructure

agile-trac
Agile Integrated Project Collaboration

clearpool.io
Real Insight, Real Liquidity

May 6 Flash Crash

yeaup *Be First To Know*

ended up at

NYSE Euronext
Powering the exchanging world.

Head of Technology

TP ICAP
Keeping Markets in Motion

Making Sense of Social Media ...

It's a New Language

Abbreviations, acronyms, emojis, emphatic spelling. Algorithms are required to **learn the meaning of non-standard strings** and new words as they appear.

Constantly Growing

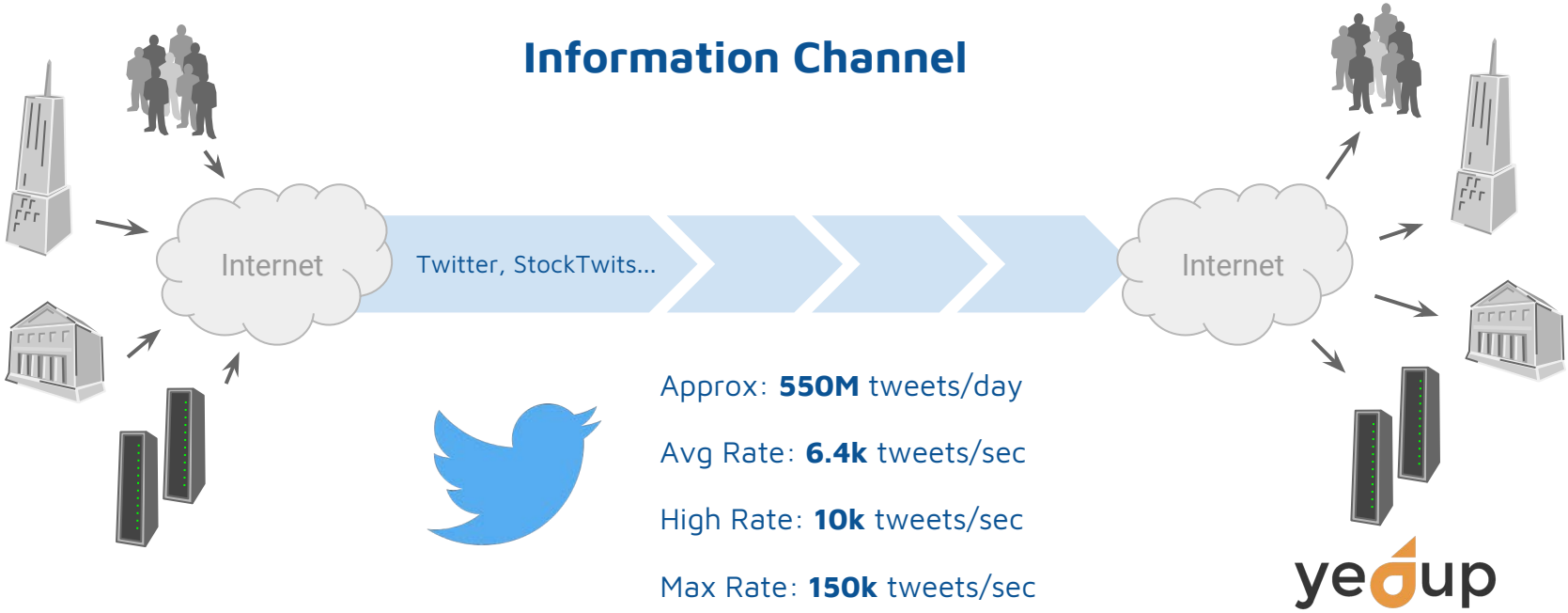
Around **1500 new words and phrases** appear in the global conversation **each day**. Over time this adds up. Words also have **different meanings in different contexts**, topic domains, and countries.

Forever Changing

Algorithms which adapt to keep pace with the way the world expresses itself are needed. These should be **context aware** and be suitable for all **domain specific** applications.

... is not easy!

Well, what is it really?



... and we want to trade on this

What we Aim For

Real-Time

Process more than 100k social media posts per second, with industry leading low latency. Always deliver results in real time.

Adaptive

Use artificial intelligence to evolve continually to reflect the fluid nature of social media expression and keep pace with the latest lingo.

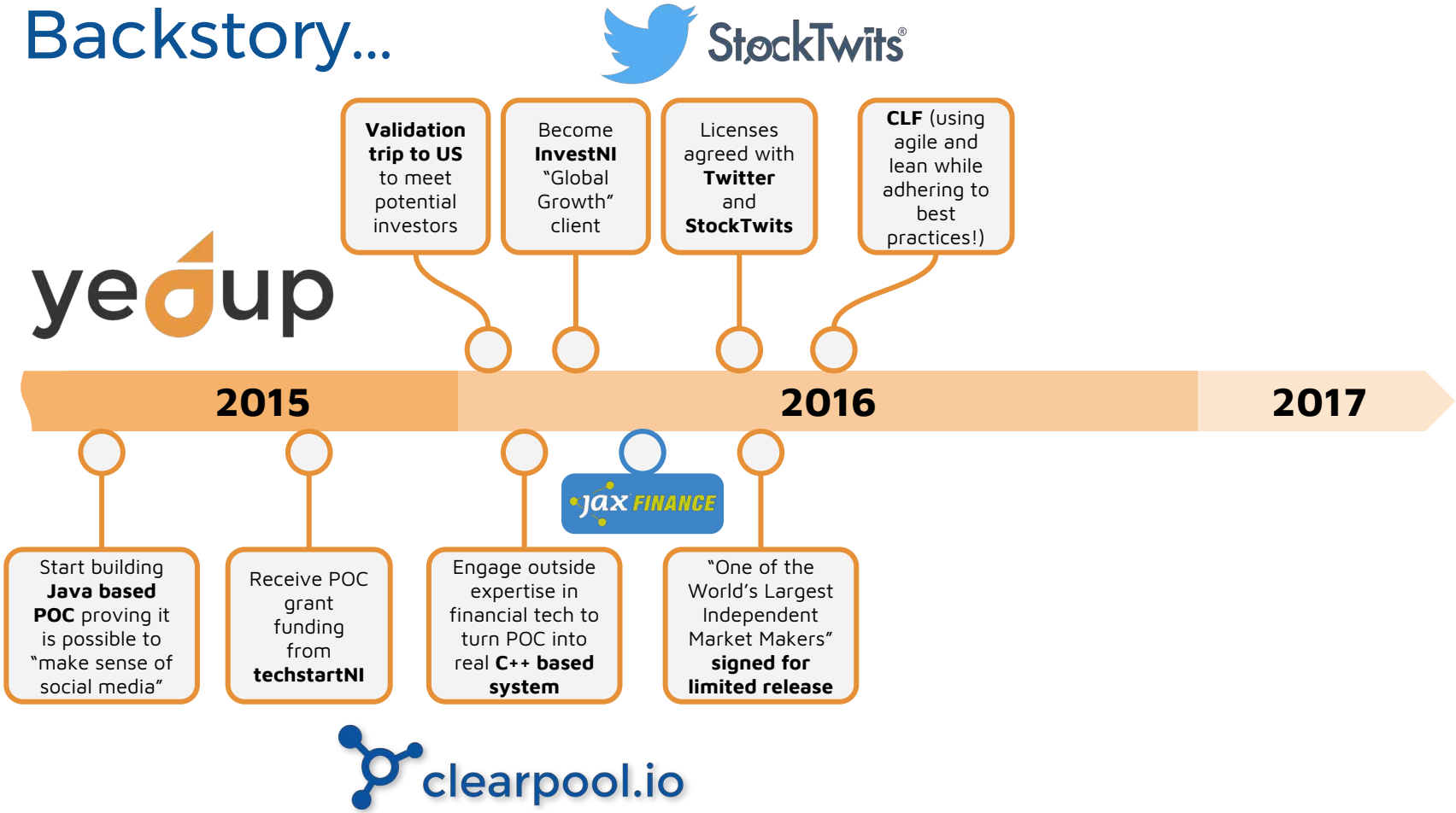
Language Agnostic

Work with all major languages and script systems. Be able to cover social media channels in Europe, Middle East, Africa, Asia and the Americas.

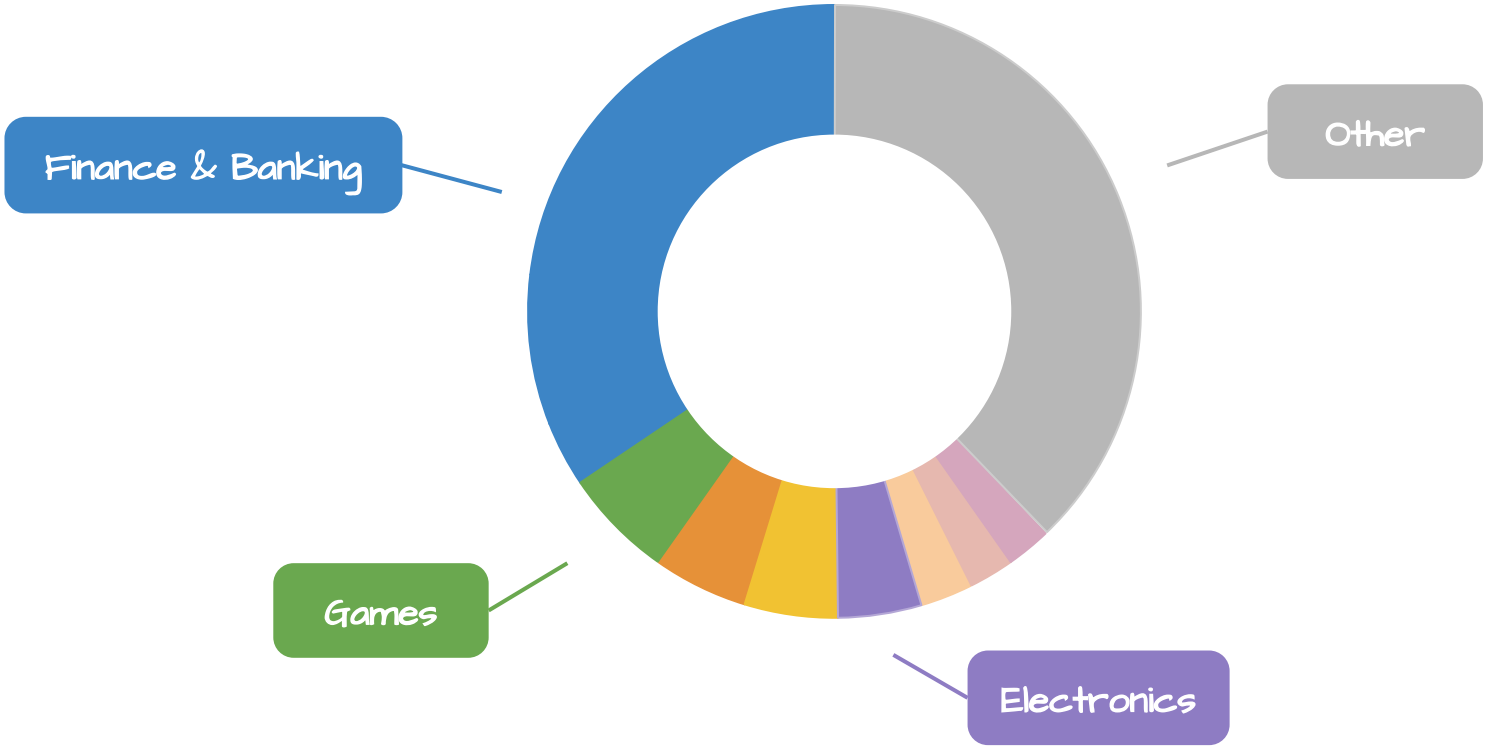
Domain Aware

Machine learning can also capture the domain-specific meanings of certain words and phrases, so the true meaning of what is said is understood in its proper context.

Backstory...


StockTwits
Validation trip to US to meet potential investors
Become InvestNI "Global Growth" client
Licenses agreed with Twitter and StockTwits
CLF (using agile and lean while adhering to best practices!)
yeoup
2015
2016
2017
Start building Java based POC proving it is possible to "make sense of social media"
Receive POC grant funding from techstartNI
Engage outside expertise in financial tech to turn POC into real C++ based system
jax FINANCE
"One of the World's Largest Independent Market Makers" signed for limited release
clearpool.io

Context — Why C++?

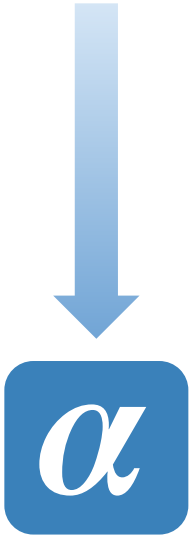


Source: http://blog.jetbrains.com/clion/2015/09/cpp-annotated-summer-edition/

Using a channel

I know what I want to hear about so I'll listen for that

I don't know exactly what I want to hear about but I'll know it when I see it



Say What?


What was said?


What was it about?

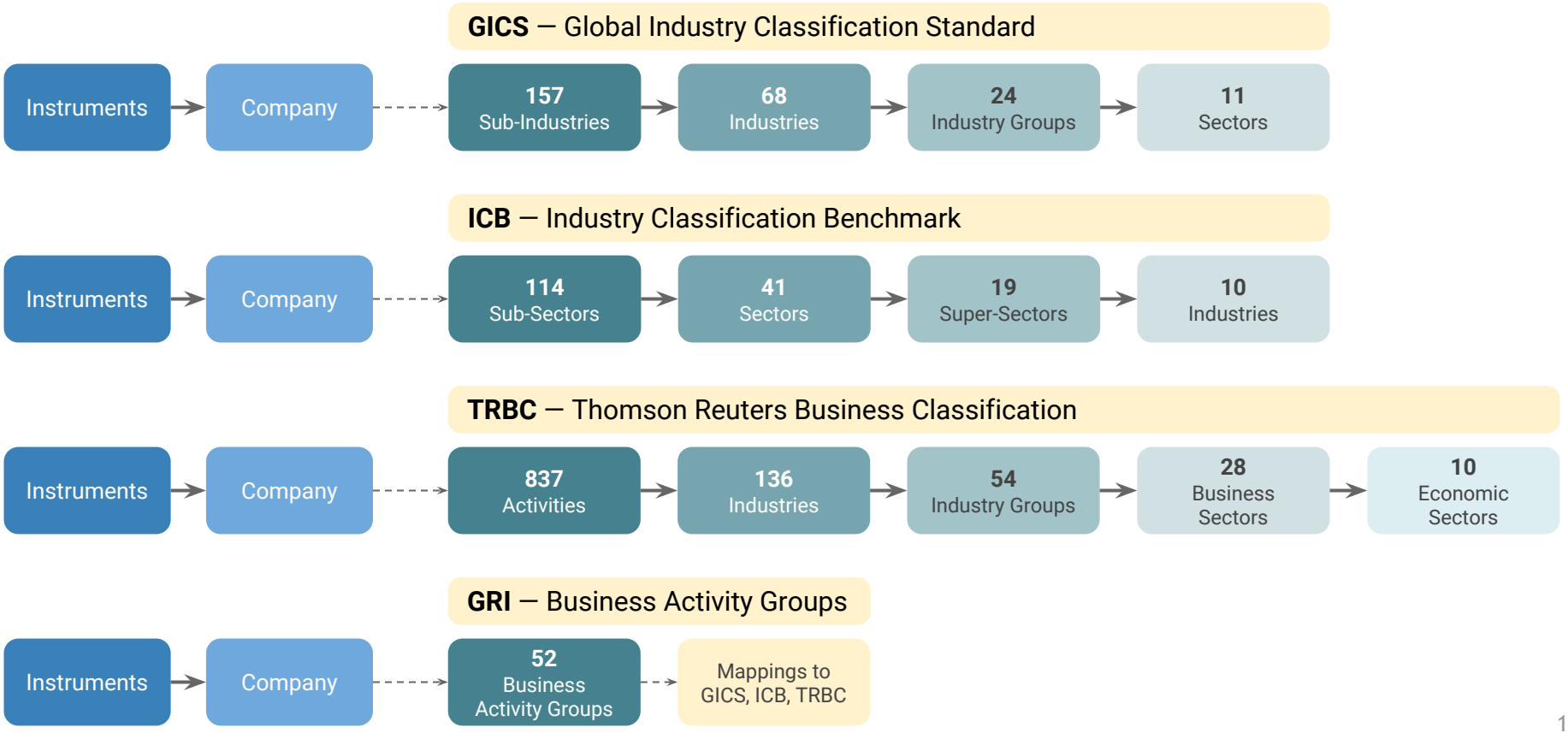

What was the opinion expressed?


Who said what was said?

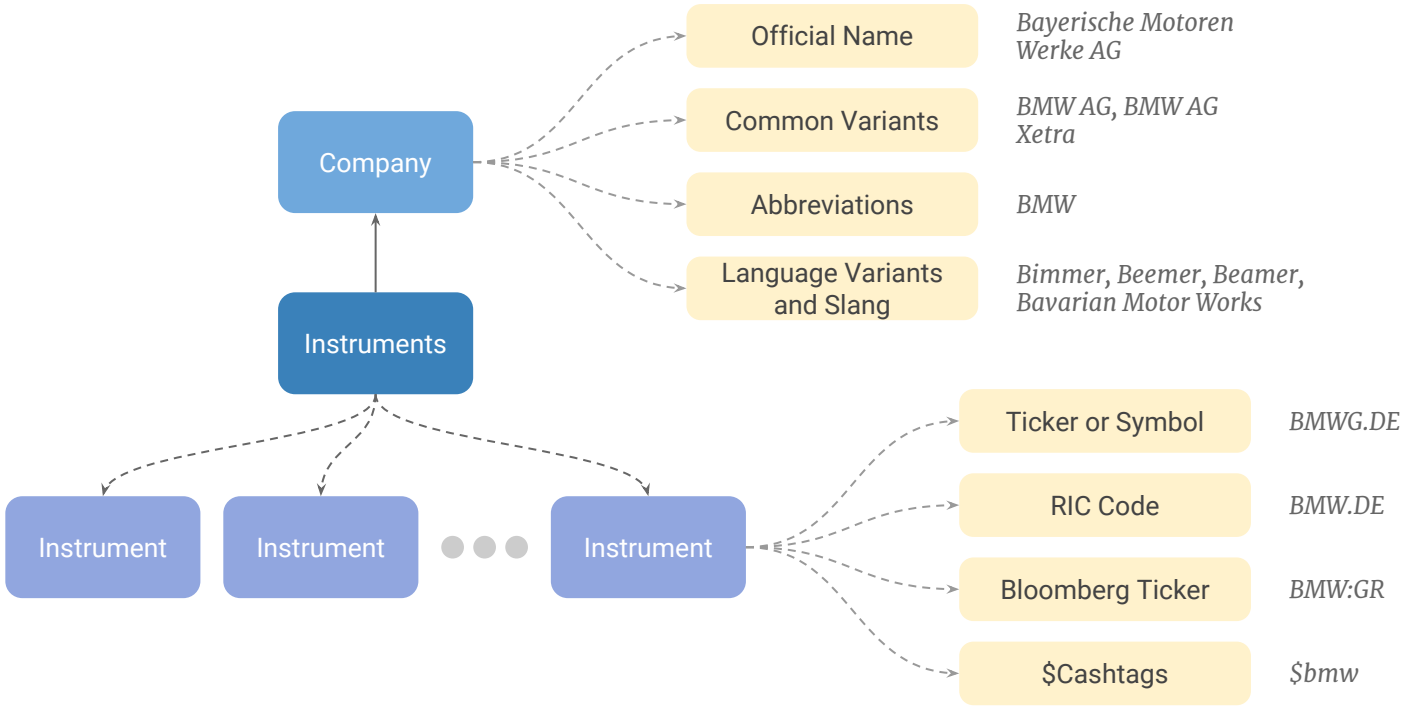

Who cared about what was said?


Has anyone said this before?

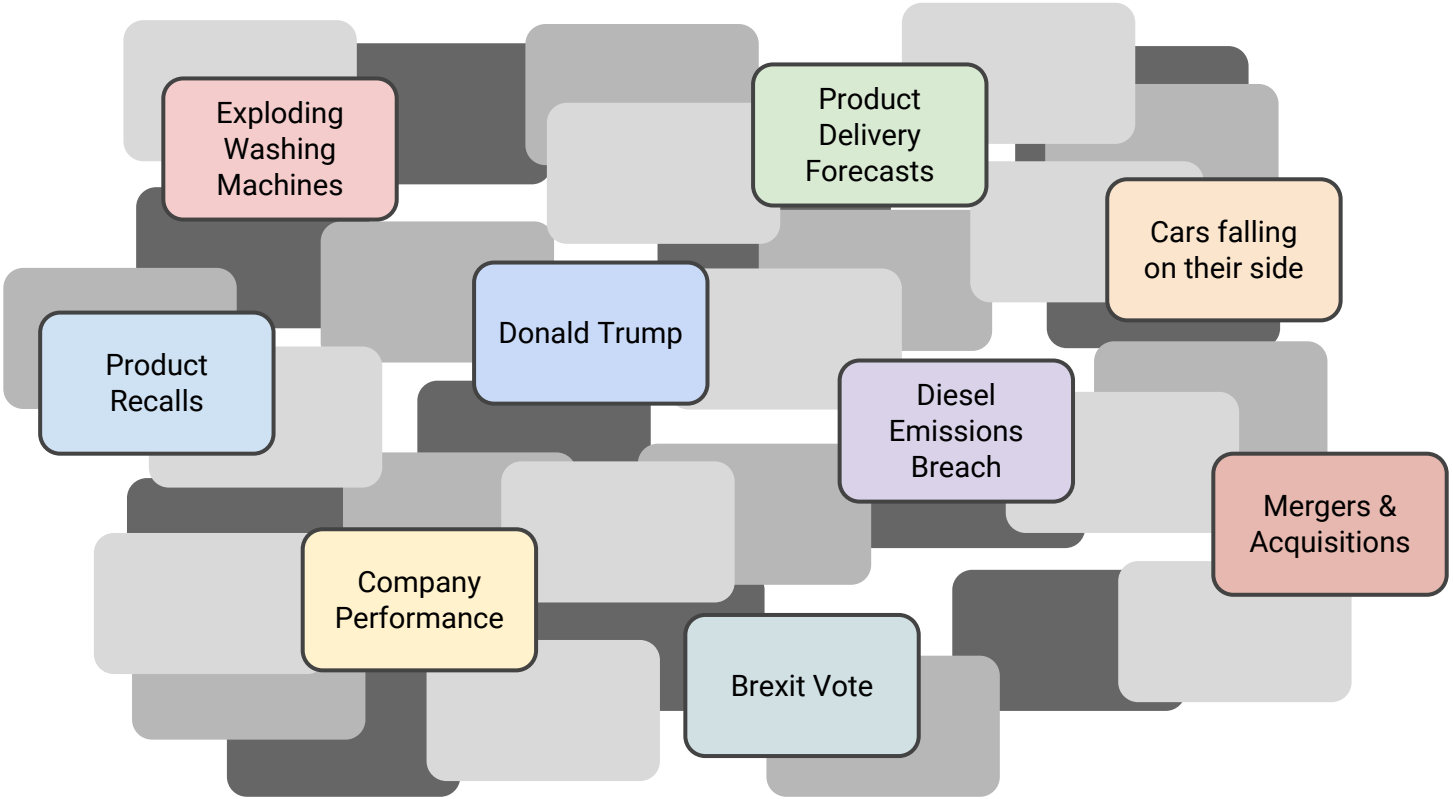
We Trade Financial Instruments



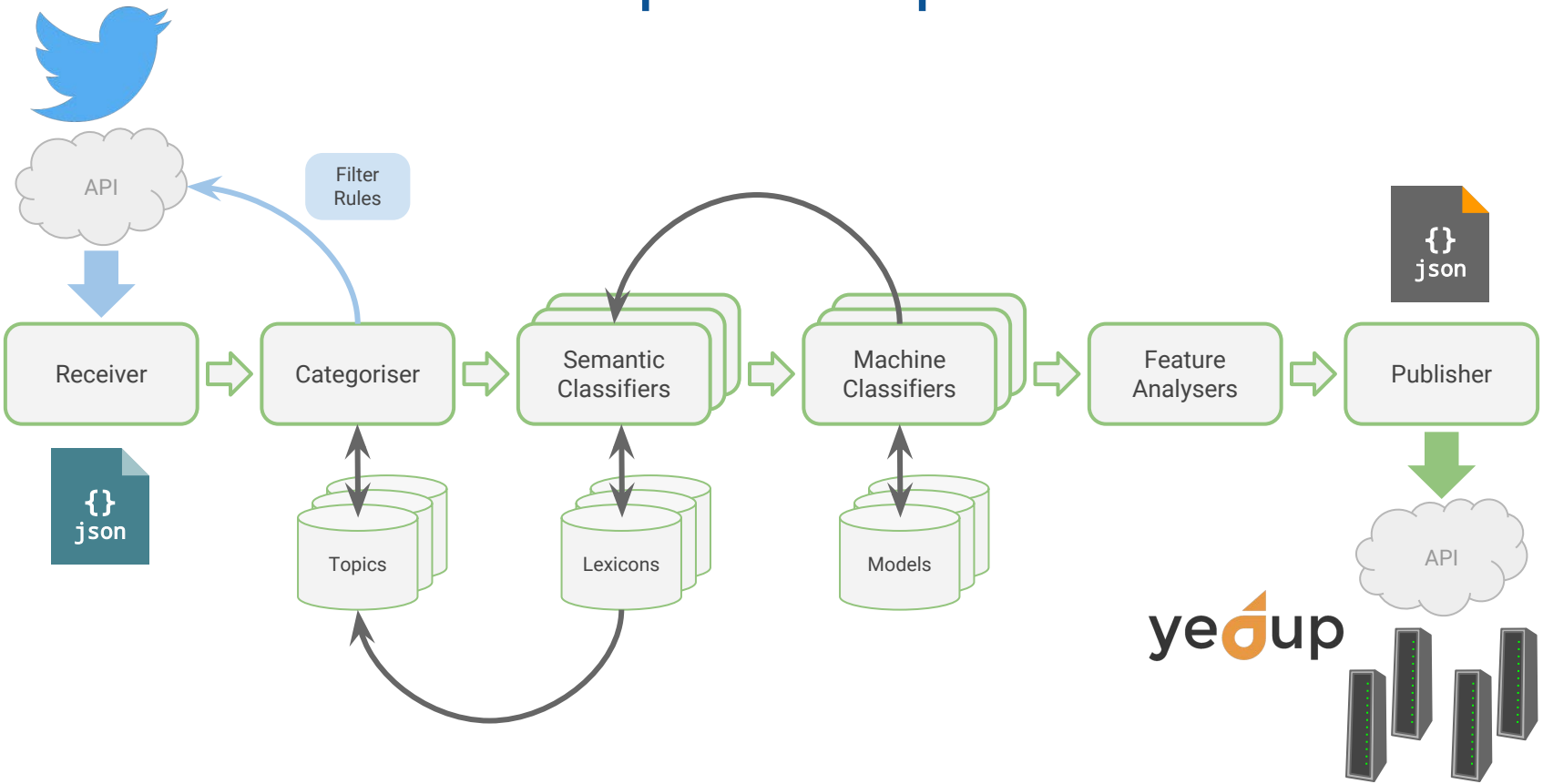
Which Instrument or Company?



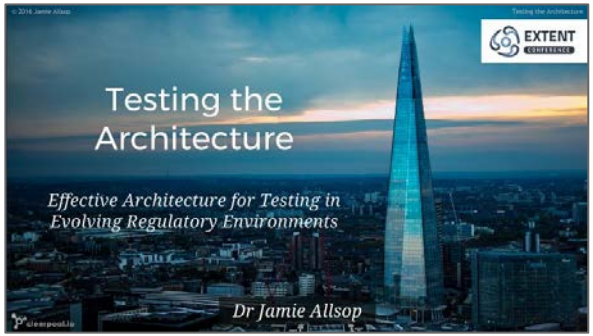
What about it?



Conceptual Pipeline



Algorithmic Architecture



Algorithmic Architecture Crash Course

This is an Architecture that

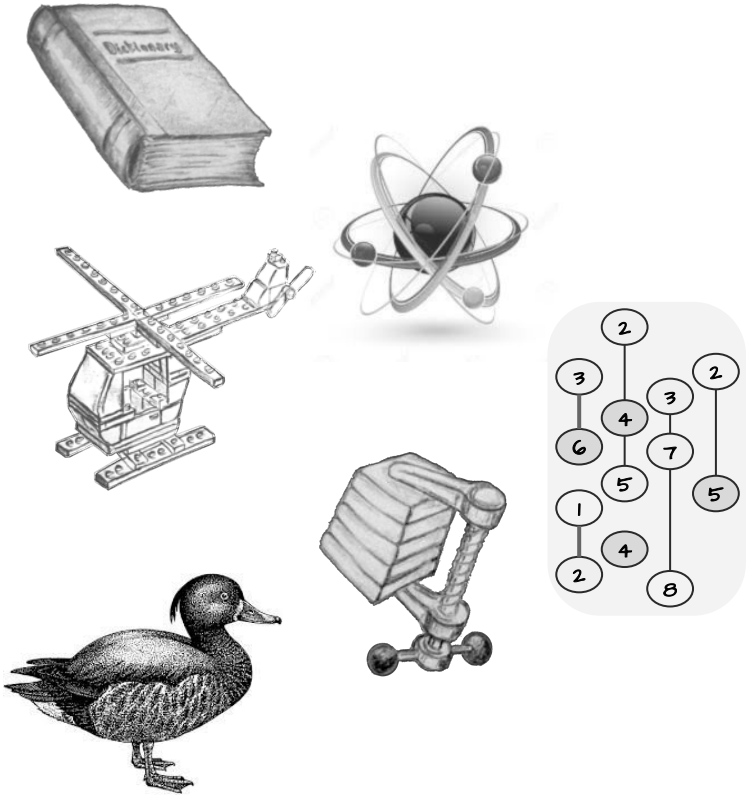
- is based on well defined building blocks
- has a clear mapping to code
- favours algorithmic optimisation over task optimisation
- allows an optimal solution
- is adaptive to a changing environment

We Achieve This By

- Exposing a Vocabulary

  that can map to code and is

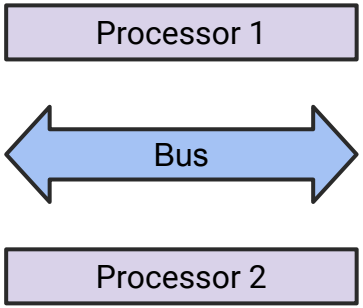
- Decomposable
- Composable
- Independently Orderable
- Compactible
- Substitutable



Towards Algorithmic Architecture

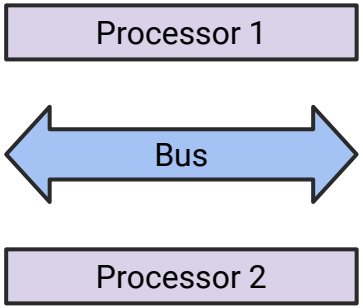
- Define **building block vocabulary elements**

```
template<class DataT>
void process( const DataT& Data );

template<class DataT>
void push( const DataT& Data );

template<class ProcessorT>
void connect( ProcessorT Processor );
```



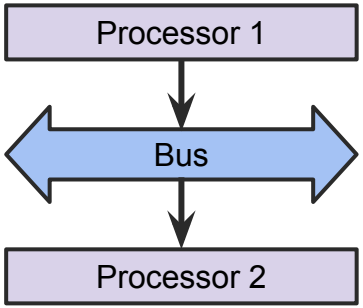
Towards Algorithmic Architecture

- Define building block vocabulary elements
- Avoid **shared state**



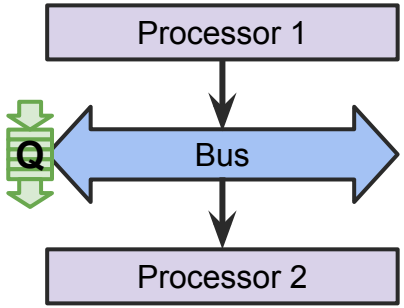
Towards Algorithmic Architecture

- Define building block vocabulary elements
- Avoid shared state
- Favour **message passing**



Towards Algorithmic Architecture

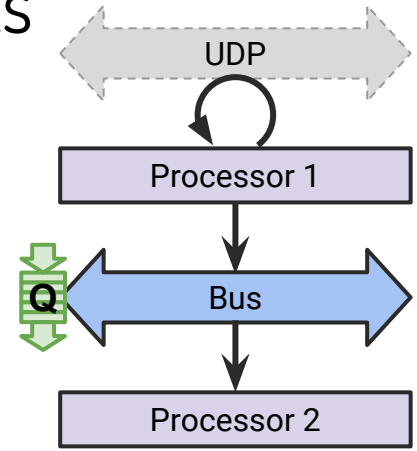
- Define building block vocabulary elements
- Avoid shared state
- Favour message passing
- Make **synchronisation points explicit** in the architecture



Synchronisation points are not composable. If you hide them you run the risk of concurrency hazards such as livelocks, starvation, deadlocks, and convoying

Towards Algorithmic Architecture

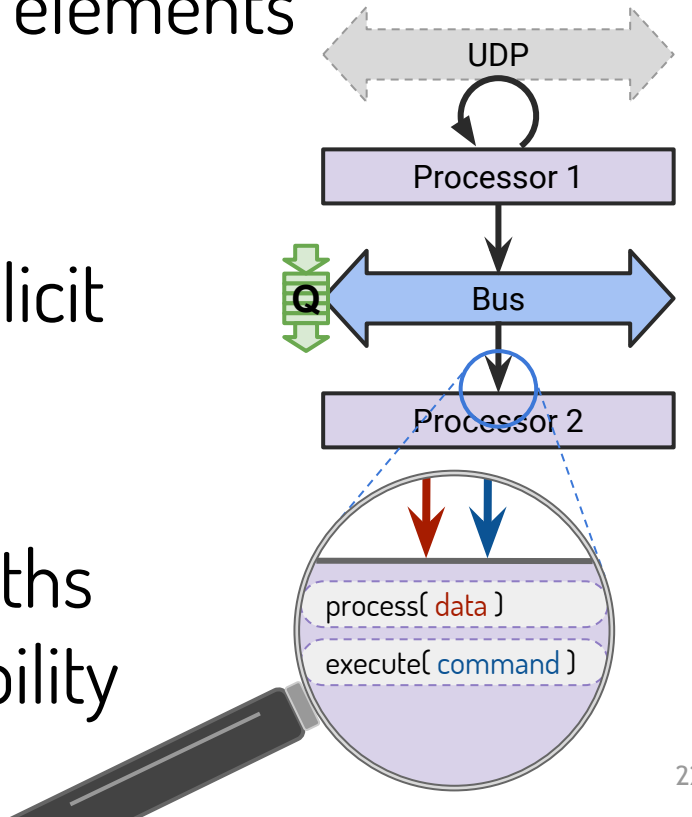
- Define building block vocabulary elements
- Avoid shared state
- Favour message passing
- Make synchronisation points explicit in the architecture
- Support **push** and **pull** models




```
enum class read_policy{ on_data, poll };
template<class ProcessorT>
void connect( ProcessorT Processor, read_policy Read );
```

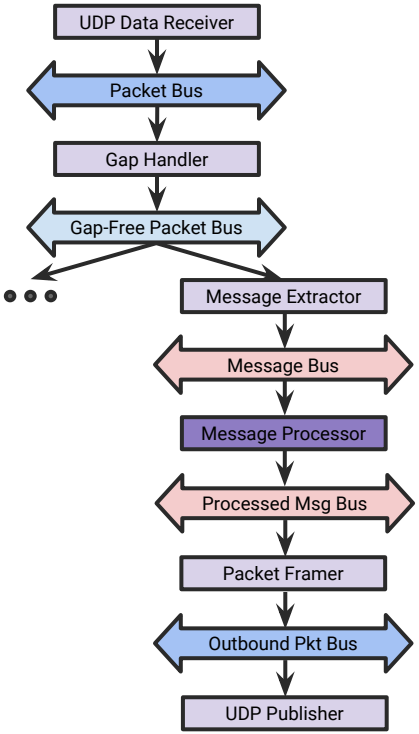
Towards Algorithmic Architecture

- Define building block vocabulary elements
- Avoid shared state
- Favour message passing
- Make synchronisation points explicit in the architecture
- Support push and pull models
- Separate Data and Command paths
- Static Polymorphism for adaptability

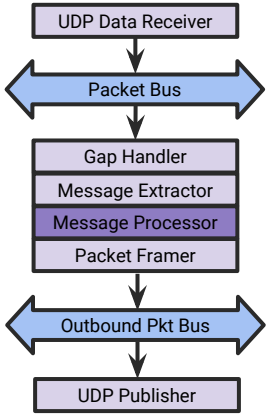


Simple Example

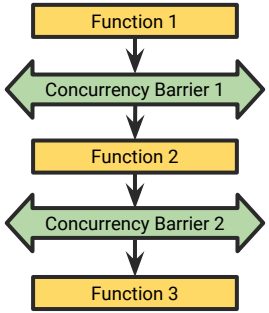
1 Design Using a Real Vocabulary of Real Components



2 Compact Architecture by removing conceptual components

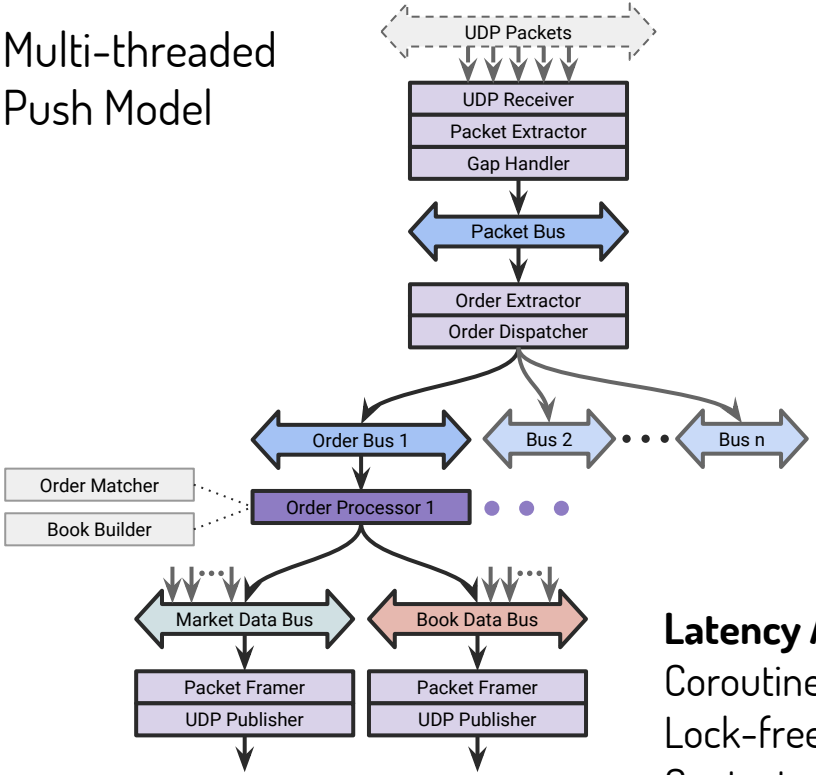


3 Compile to Optimised Implementation with zero abstraction cost

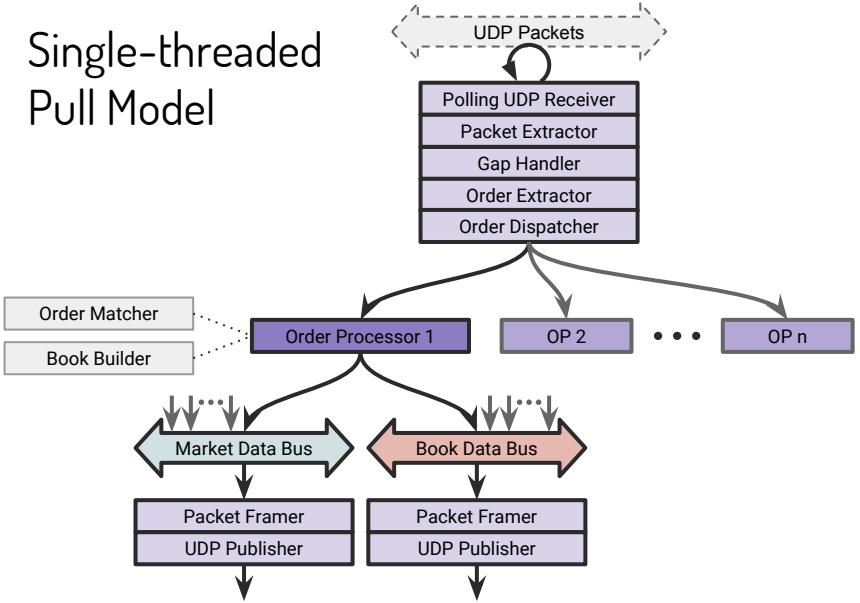


Different Performance Trade-offs

Multi-threaded
Push Model



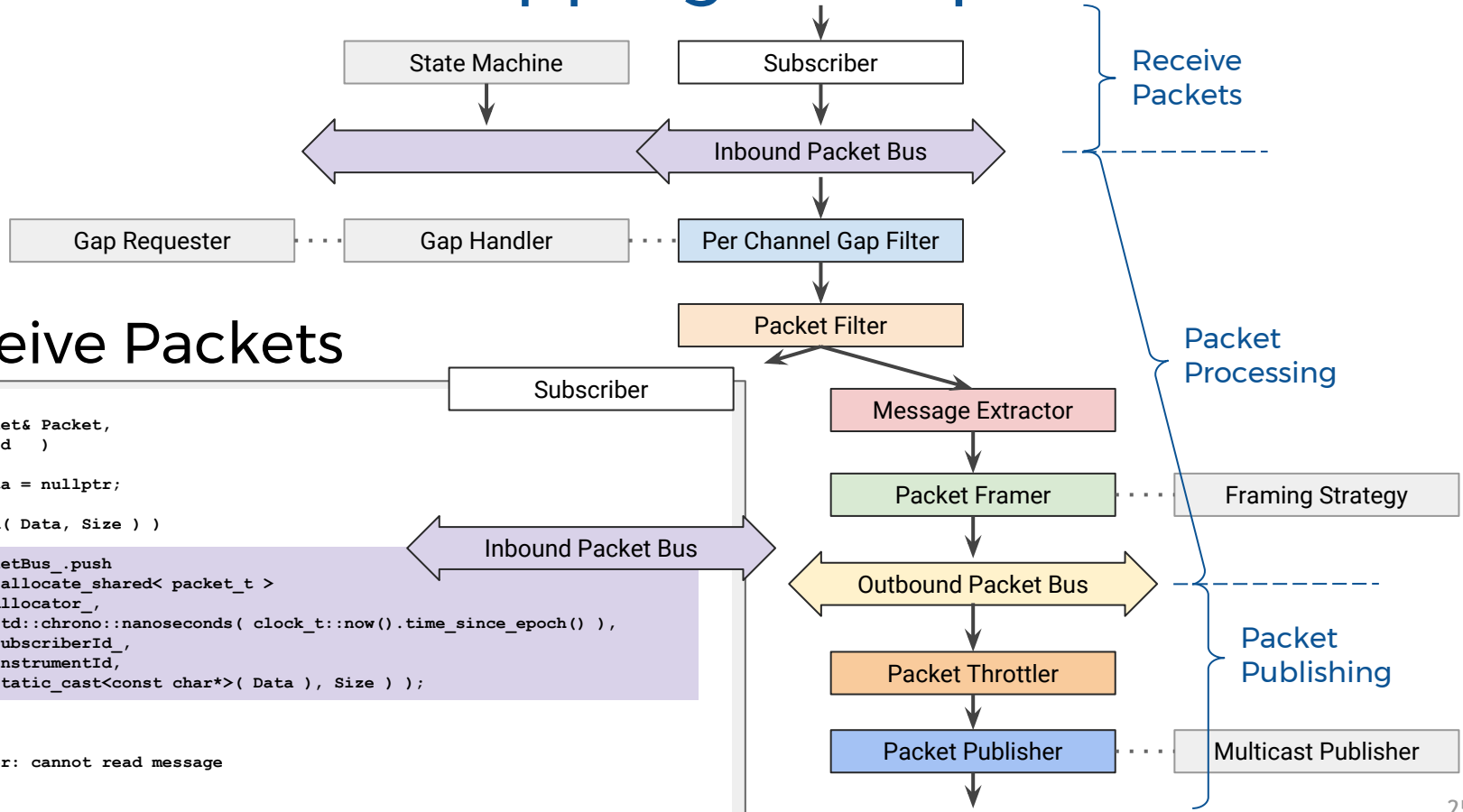
Single-threaded
Pull Model



Latency Agnostic
Coroutines?
Lock-free Queues?
Context-switches?

Scaling Agnostic
Single Process → Multiple Processes?
Single Core → Multiple Cores?
Single Server → Multiple Servers?

Code Mapping Example



Receive Packets

```
void on_packet
(   const data_packet& Packet,
    int InstrumentId   )
{
    const void* Data = nullptr;
    size_t Size;
    if( Packet.read( Data, Size ) )
    {
        InboundPacketBus_.push
            ( std::allocate_shared< packet_t >
                ( Allocator_,
                  std::chrono::nanoseconds( clock_t::now().time_since_epoch() ),
                  SubscriberId_,
                  InstrumentId,
                  static_cast<const char*>( Data ), Size ) );
    }
    else
    {
        // log error: cannot read message
    }
}
```

Packet Processing

```
void process( const shared_inbound_packet& InboundPacket )
{
    if( InboundPacket->seq_num() == ExpectedSeqNum )
    {
        ExpectedSeqNum = InboundPacket->seq_num() + InboundPacket->header().num_msgs();
        GapHandler_.update_expected_seq_num( ExpectedSeqNum, ChannelId );

        if(     InboundPacket->header().num_msgs()
            &&  InboundPacket->header().delivery_flag() == format::delivery_flag::original_message )
        {
            while( shared_message_t Message = InboundPacket->pop_front() )
            {
                if( FramingStrategy_->incoming_message_triggers_send( OutboundPacket_->size(), Message->size() ) )
                {
                    SeqNum_ += NumMsgsInPrevPacket_;
                    LastFrameTime_ = clock_t::now().time_since_epoch();
                    OutboundPacket_->assign_seq_num( SeqNum_ );
                    OutboundPacketBus_->push( OutboundPacket_ );
                    NumMsgsInPrevPacket_ = OutboundPacket_->header().num_msgs();
                    OutboundPacket_ = std::make_shared<outbound_packet_t>( format::delivery_flag::original_message );
                }
                OutboundPacket_->push_back( Message );
                if( FramingStrategy_->packet_requires_immediate_send( OutboundPacket_->size(), Message->last_message_in_packet() ) )
                {
                    SeqNum_ += NumMsgsInPrevPacket_;
                    LastFrameTime_ = clock_t::now().time_since_epoch();
                    OutboundPacket_->assign_seq_num( SeqNum_ );
                    OutboundPacketBus_->push( OutboundPacket_ );
                    NumMsgsInPrevPacket_ = OutboundPacket_->header().num_msgs();
                    OutboundPacket_ = std::make_shared<outbound_packet_t>( format::delivery_flag::original_message );
                }
            }
        }
        else
        {
            // send command::category::notification - packet_discarded
        }
    }
    else if( InboundPacket->seq_num() > ExpectedSeqNum )
    {
        ExpectedSeqNum = GapHandler_.handle_unexpected_packet( InboundPacket, ExpectedSeqNum, ChannelId );
    }
    else if( InboundPacket->seq_num() < ExpectedSeqNum )
    {
        // log and ignore
    }
}
```

Inbound Packet Bus

Per Channel Gap Filter

Packet Filter

Message Extractor

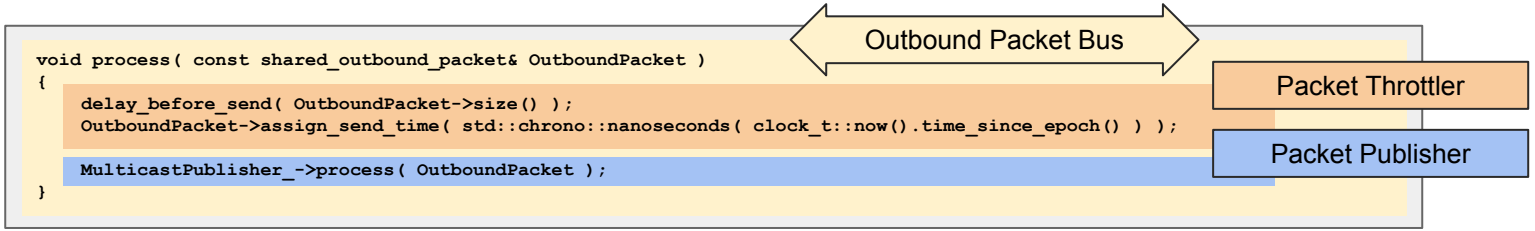
Packet Framer

Outbound Packet Bus

Outbound Packet Bus

Lastly…

Publish Packets

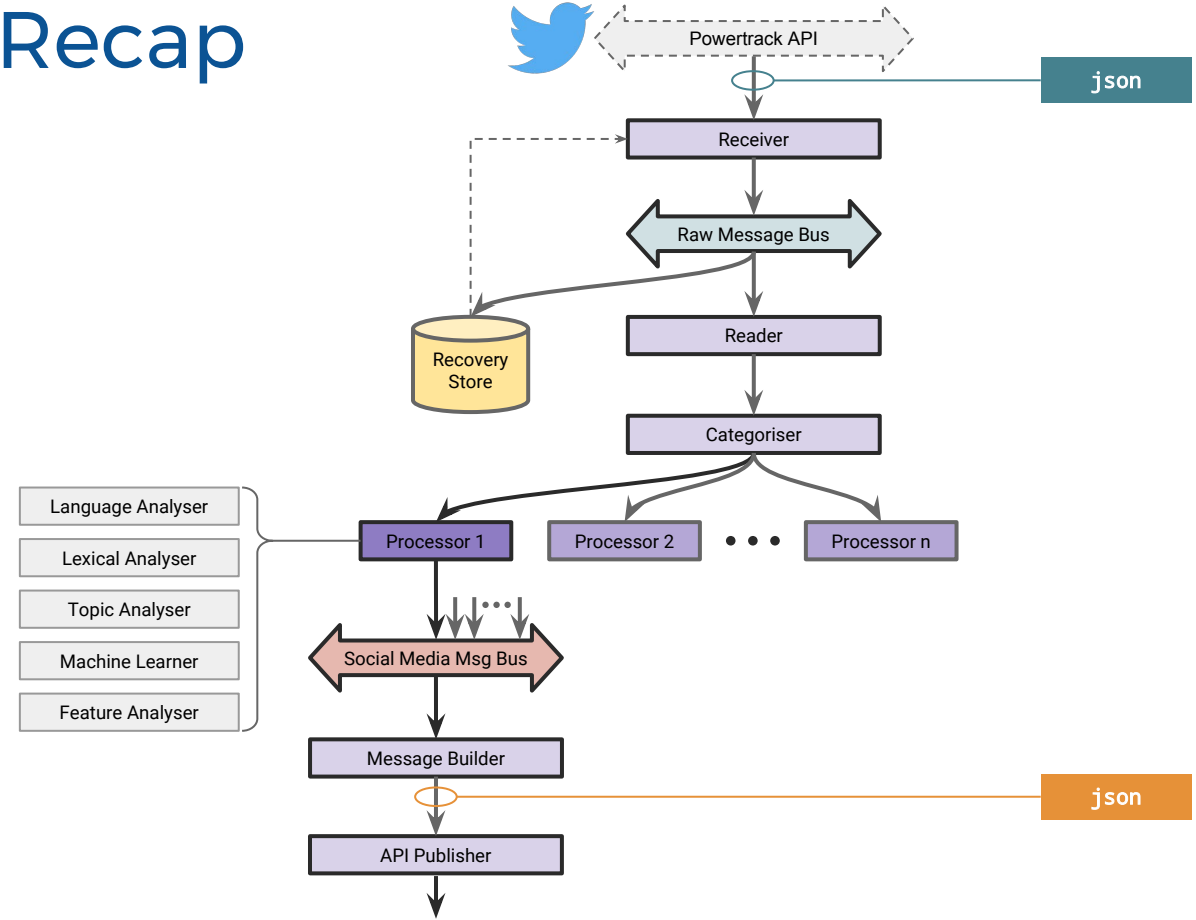


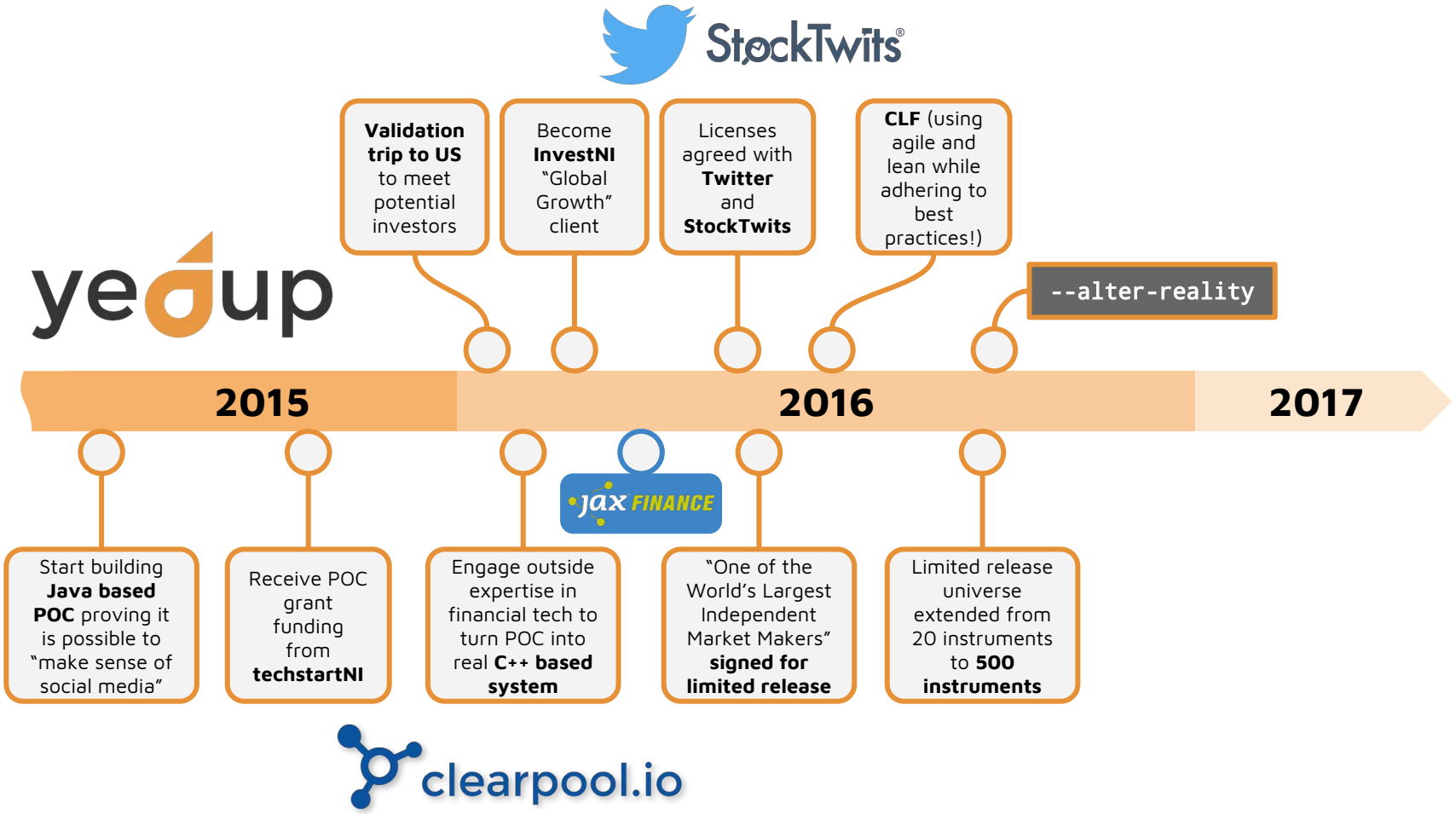

```
void process( const shared_outbound_packet& OutboundPacket )
{
    delay_before_send( OutboundPacket->size() );
    OutboundPacket->assign_send_time( std::chrono::nanoseconds( clock_t::now().time_since_epoch() ) );

    MulticastPublisher_->process( OutboundPacket );
}
```

Vocabulary elements map directly to code

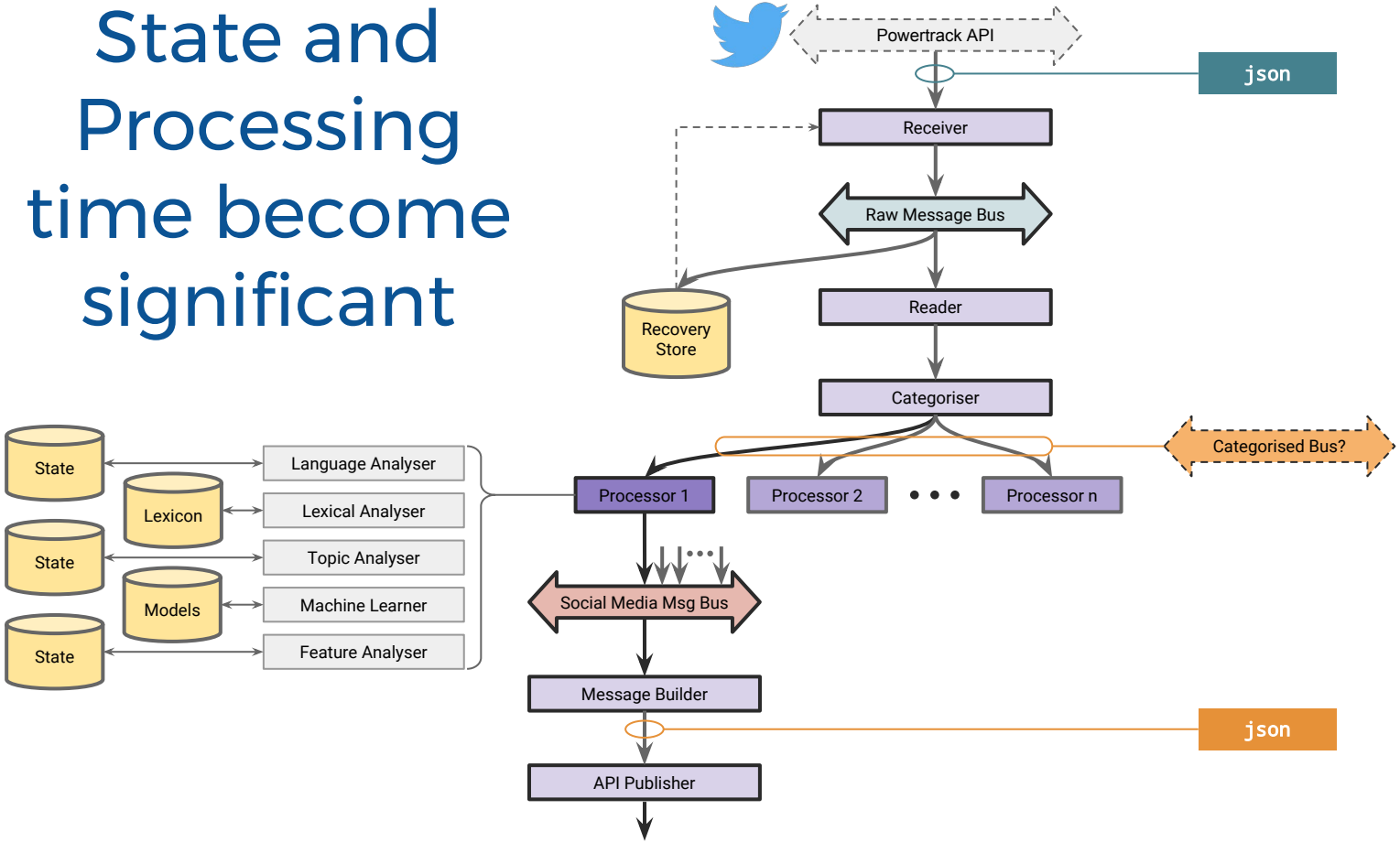
- Code still lives in separate ‘modules’
- Maintained and tested separately
- Communication through building block interfaces
- Abstraction cost removed but clarity retained
- Easy to change, fix, replace

Let's Recap

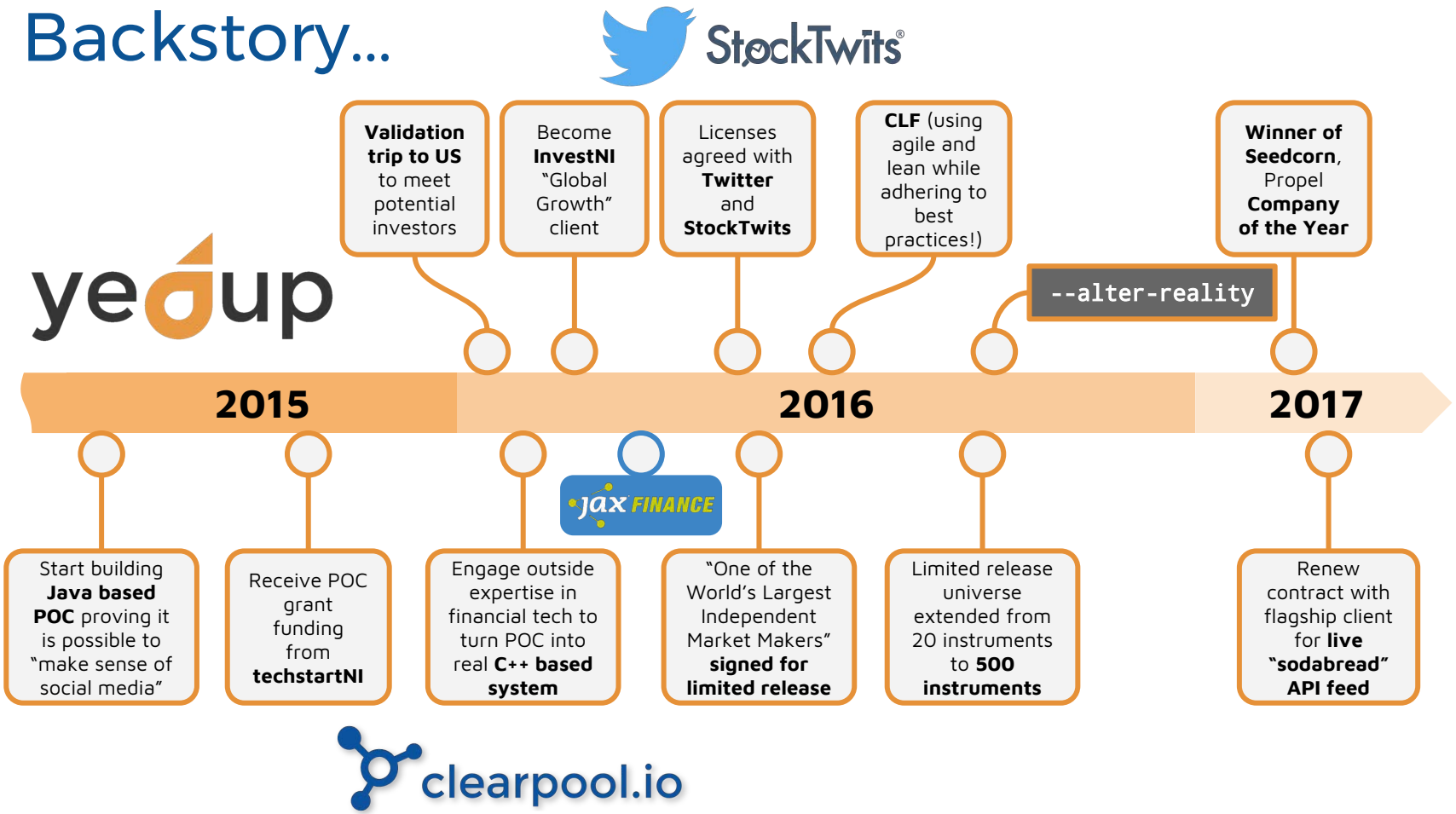

Powertrack API
json
Receiver
Raw Message Bus
Recovery Store
Reader
Categoriser
Language Analyser
Lexical Analyser
Topic Analyser
Machine Learner
Feature Analyser
Processor 1
Processor 2
Processor n
Social Media Msg Bus
Message Builder
json
API Publisher


StockTwits
Validation trip to US to meet potential investors
Become InvestNI "Global Growth" client
Licenses agreed with Twitter and StockTwits
CLF (using agile and lean while adhering to best practices!)
--alter-reality
yeoup
2015
2016
2017
Start building Java based POC proving it is possible to "make sense of social media"
Receive POC grant funding from techstartNI
Engage outside expertise in financial tech to turn POC into real C++ based system
jax FINANCE
"One of the World's Largest Independent Market Makers" signed for limited release
Limited release universe extended from 20 instruments to 500 instruments
clearpool.io

State and Processing time become significant

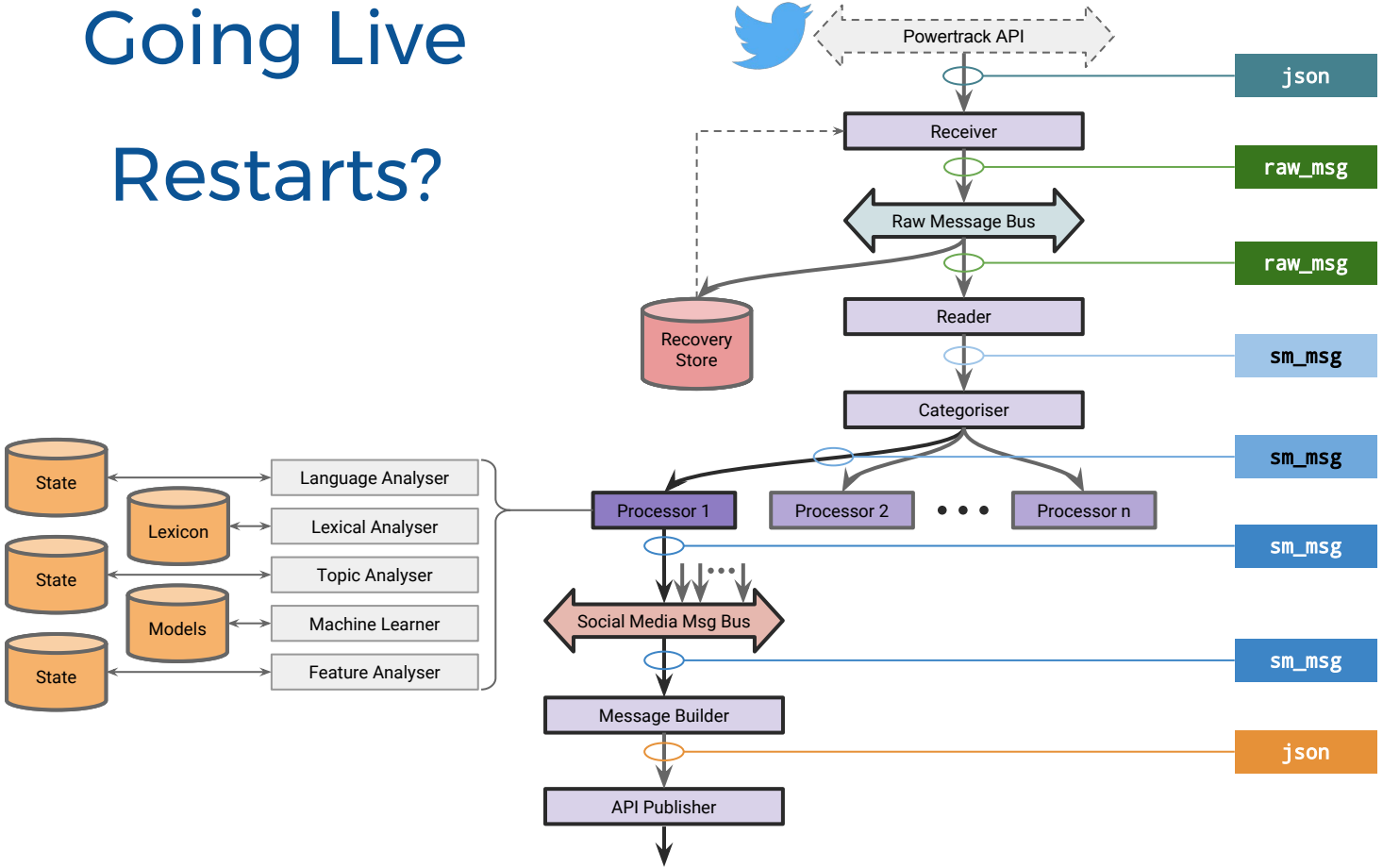


Backstory...


StockTwits
yeoup
2015
2016
2017
Validation trip to US to meet potential investors
Become InvestNI "Global Growth" client
Licenses agreed with Twitter and StockTwits
CLF (using agile and lean while adhering to best practices!)
--alter-reality
Winner of Seedcorn, Propel Company of the Year
Start building Java based POC proving it is possible to "make sense of social media"
Receive POC grant funding from techstartNI
Engage outside expertise in financial tech to turn POC into real C++ based system
jax FINANCE
"One of the World's Largest Independent Market Makers" signed for limited release
Limited release universe extended from 20 instruments to 500 instruments
Renew contract with flagship client for live "sodabread" API feed
clearpool.io

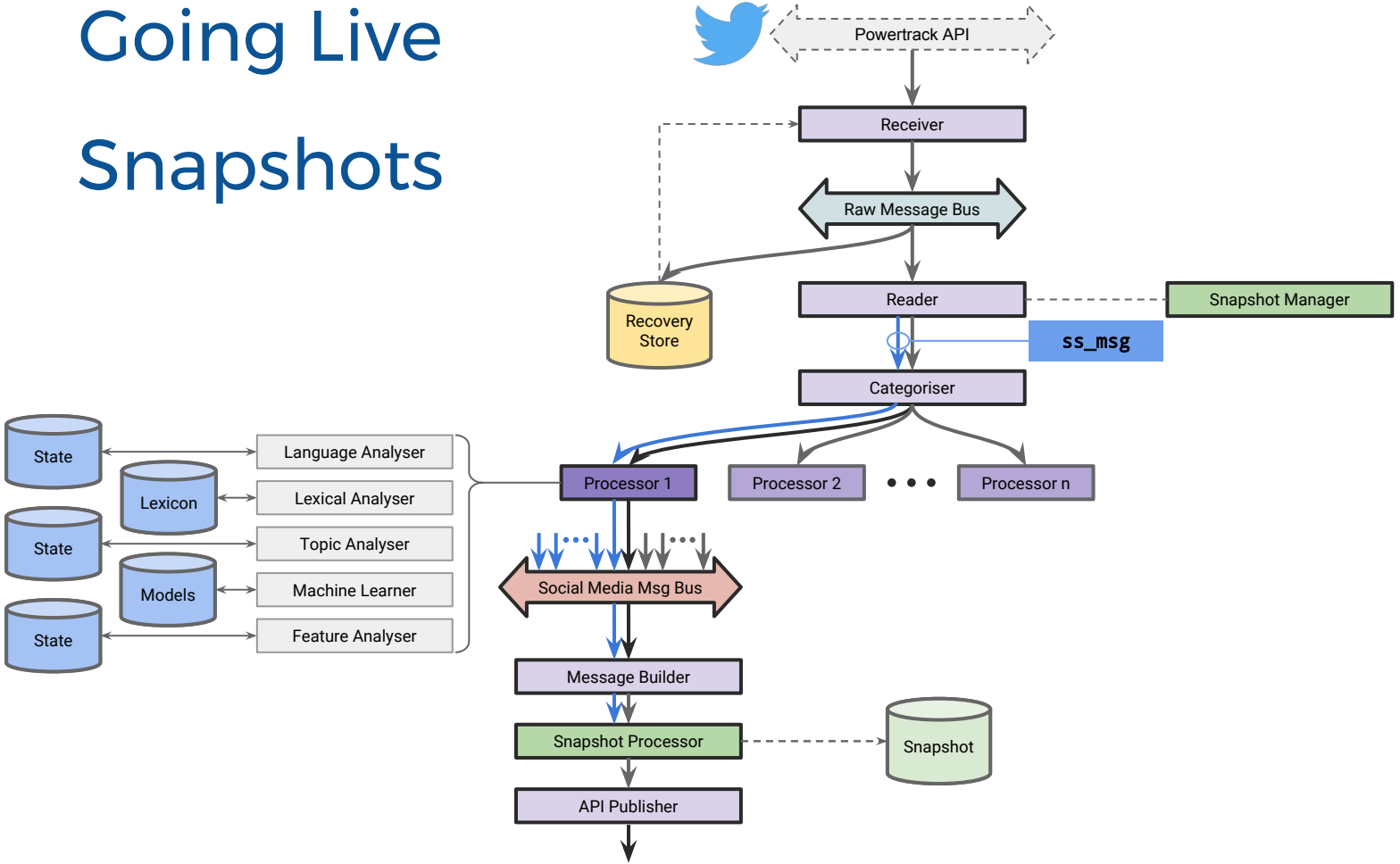
Going Live

Restarts?



Going Live

Snapshots



Getting there...

StockTwits

yeoup

Validation trip to US to meet potential investors

Become **InvestNI** "Global Growth" client

Licenses agreed with **Twitter** and **StockTwits**

CLF (using agile and lean while adhering to best practices!)

`--alter-reality`

Winner of Seedcorn, Propel **Company of the Year**

Live API Feed being used

2015 | **2016** | **2017**

Start building **Java based POC** proving it is possible to "make sense of social media"

Receive POC grant funding from **techstartNI**

Engage outside expertise in financial tech to turn POC into real **C++ based system**

jax FINANCE

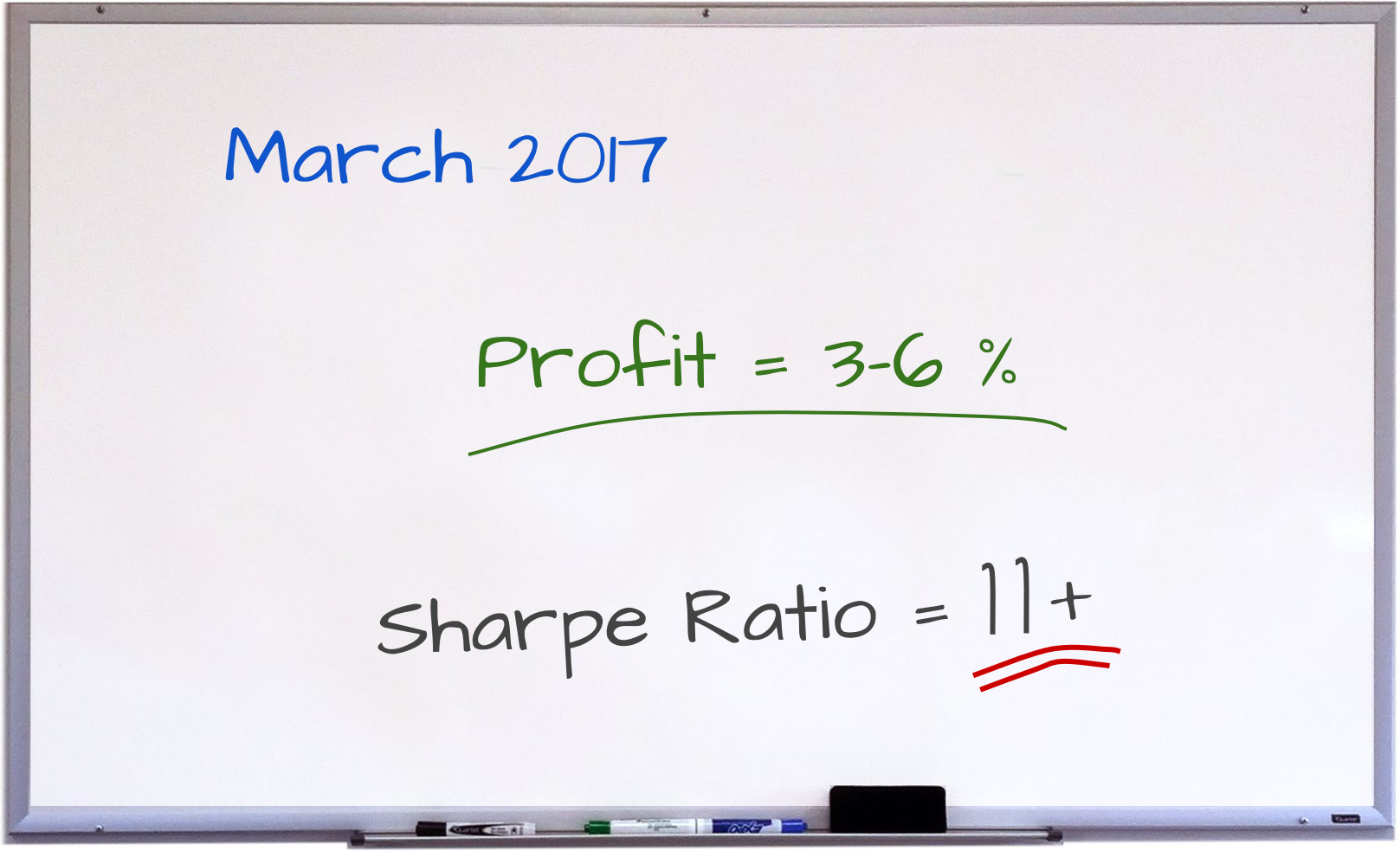
"One of the World's Largest Independent Market Makers" **signed for limited release**

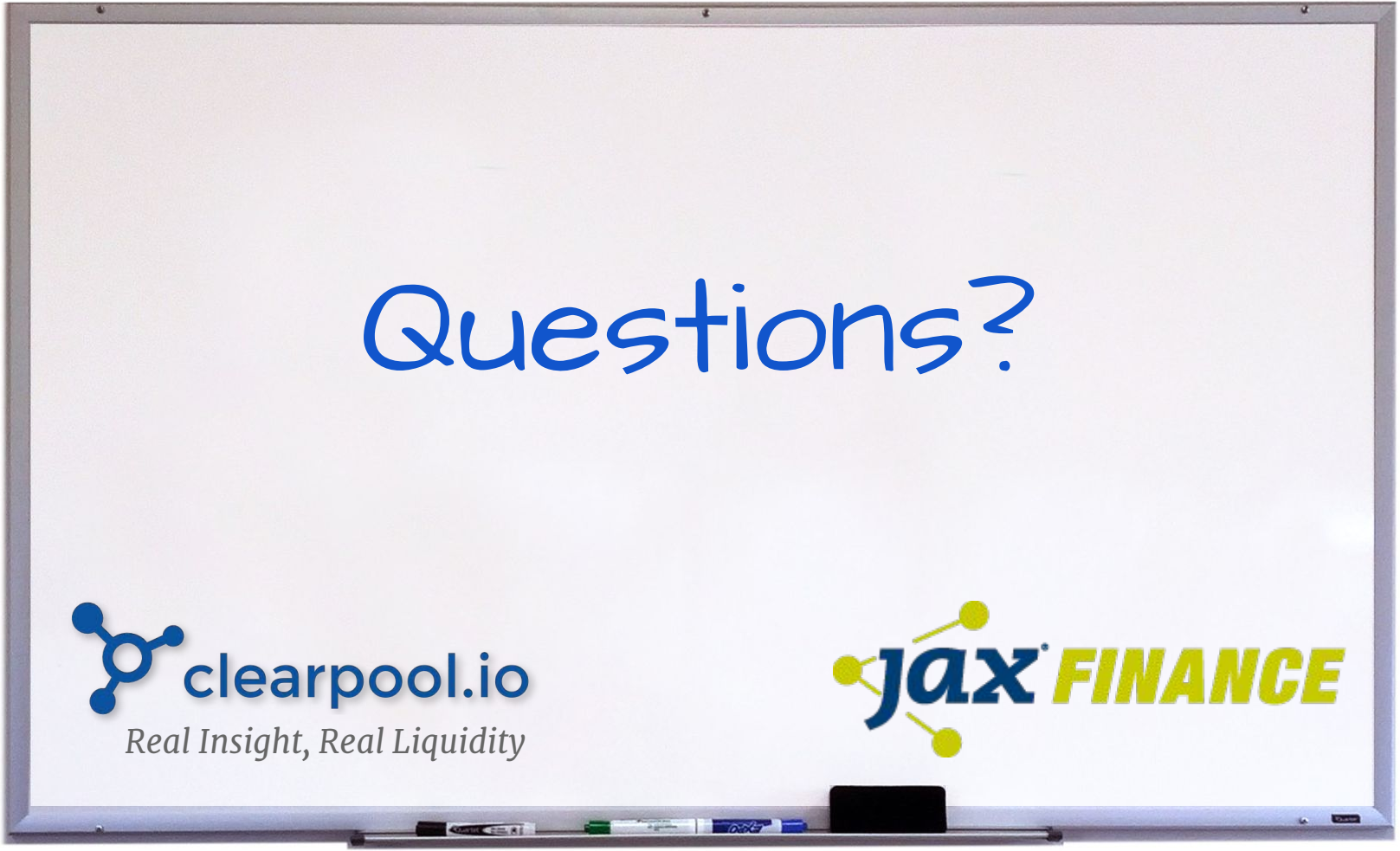
Limited release universe extended from 20 instruments to **500 instruments**

Renew contract with flagship client for **live "sodabread" API feed**

jax FINANCE

clearpool.io


March 2017
Profit = 3-6 %
Sharpe Ratio = 11+


Questions?
clearpool.io
Real Insight, Real Liquidity
JAX FINANCE