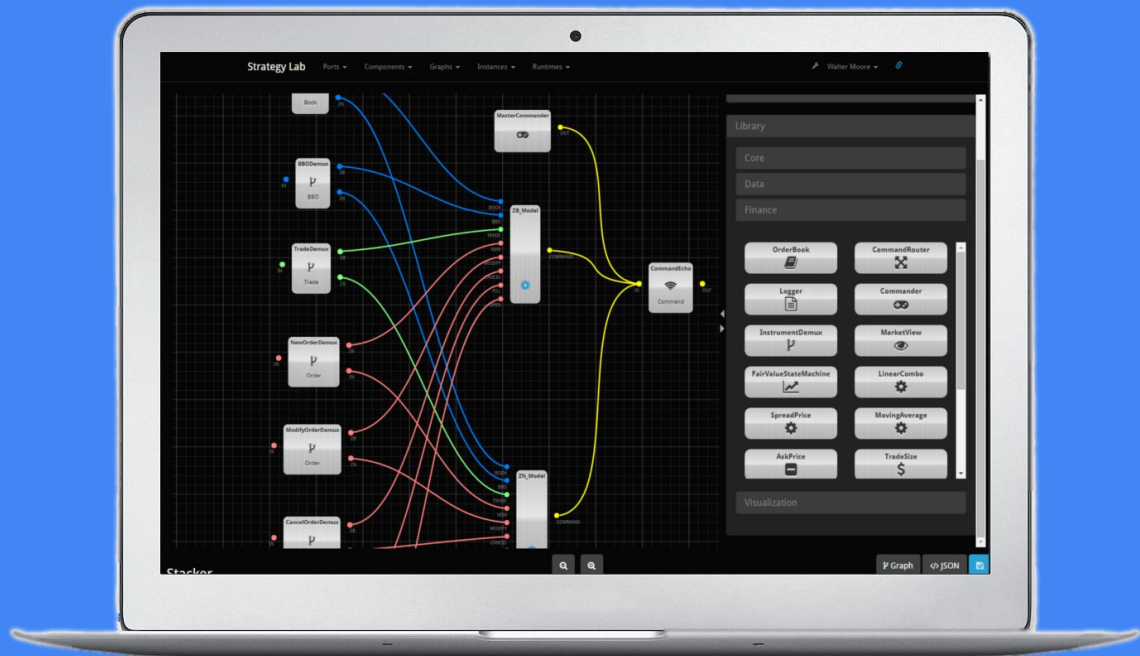
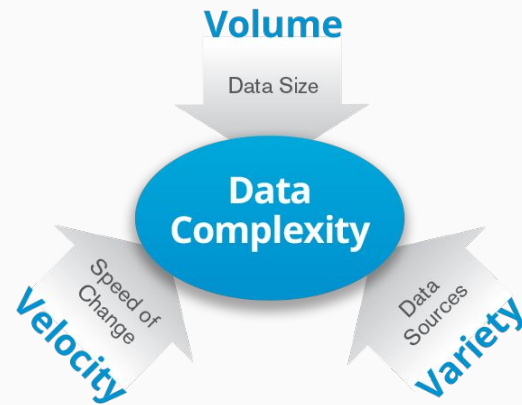


STRATEGYLAB

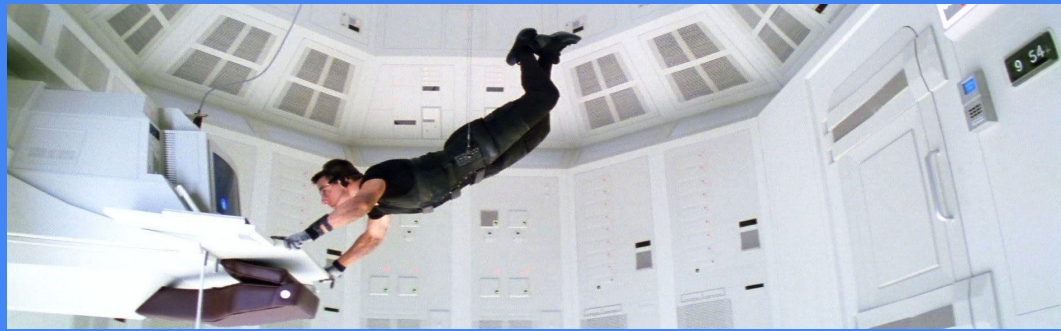


From Big Data To Big Ideas

StrategyLab is a component based graphical programming cloud PAAS for rapidly building, deploying, and managing high performance, modular and decoupled “big data” processing, analytics and visualization systems.

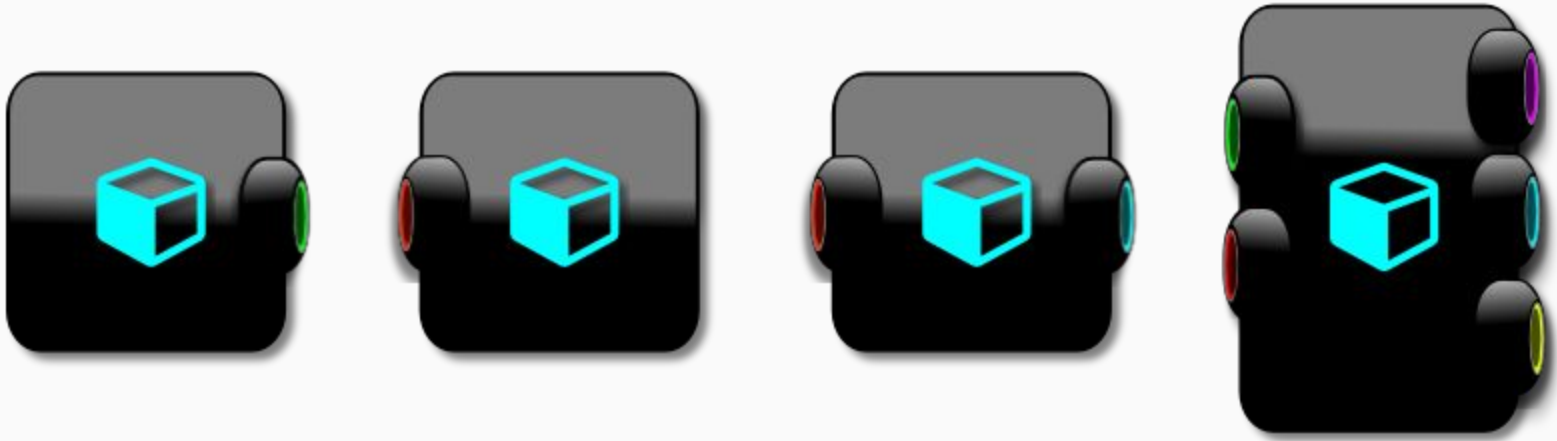


Mission



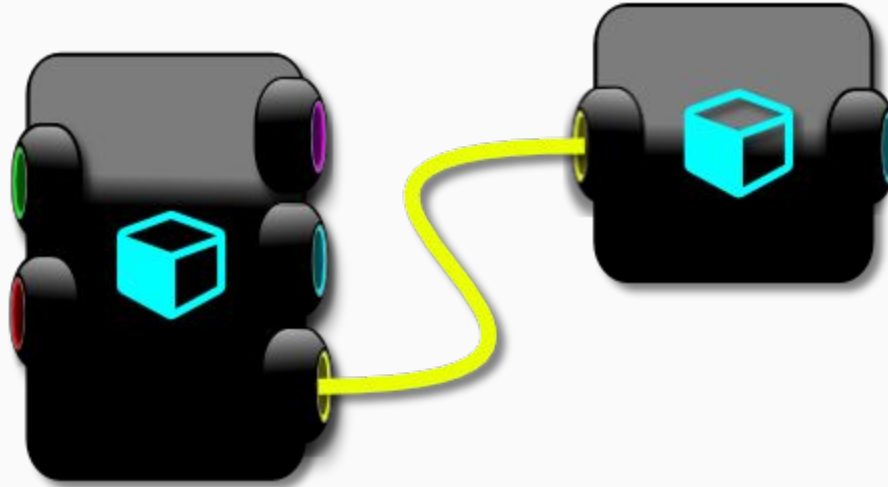
StrategyLab seeks to change the approach to big data strategy development by bridging the gap between engineering and configuration, consumption and management of a system's components. By writing components for StrategyLab, software engineers are able to rapidly deliver high performance server and user interface functionality simultaneously while maximizing code reuse. Our end users can rapidly assemble, deploy and interact with these components into connected data flow graphs, and move them to any physical runtime in the world almost instantly.

StrategyLab is all about components!



Components can have zero or more InPorts and OutPorts.

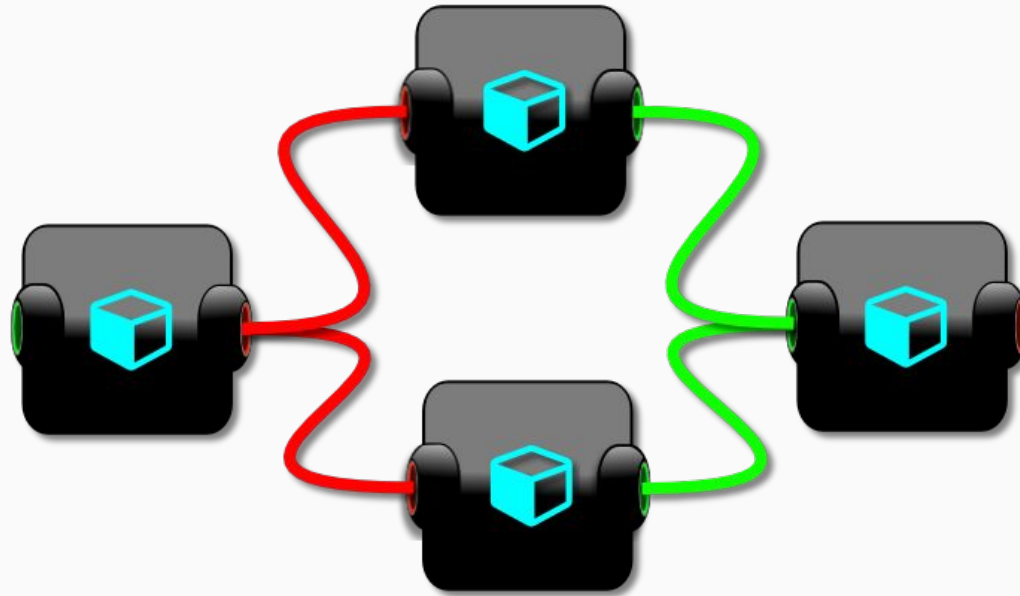
Connections



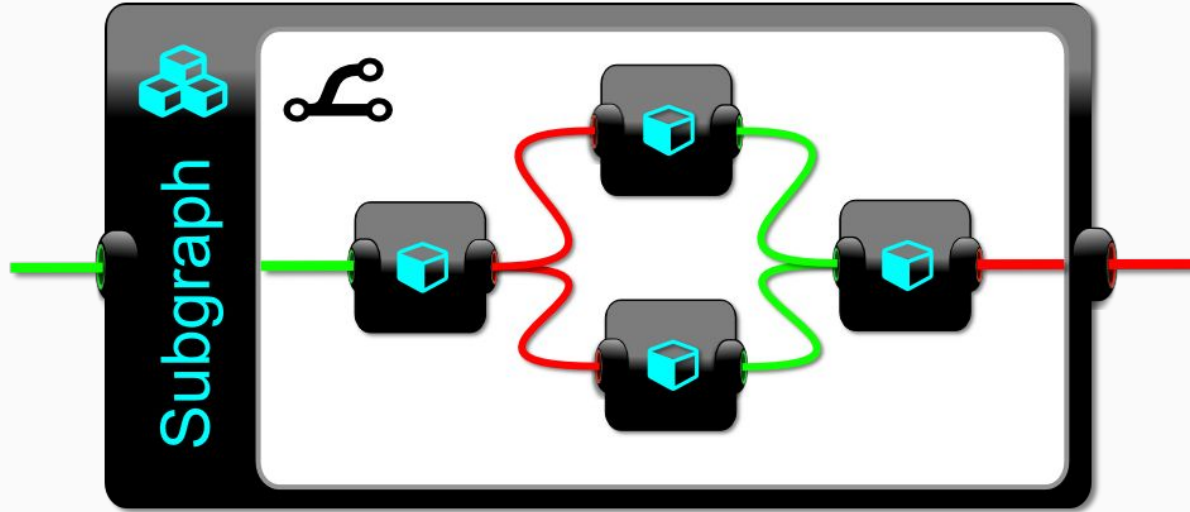
Any OutPort can be connected to any InPort of the same type.
(represented here by color)

What Is A Graph?

A graph is a set of interconnected components.

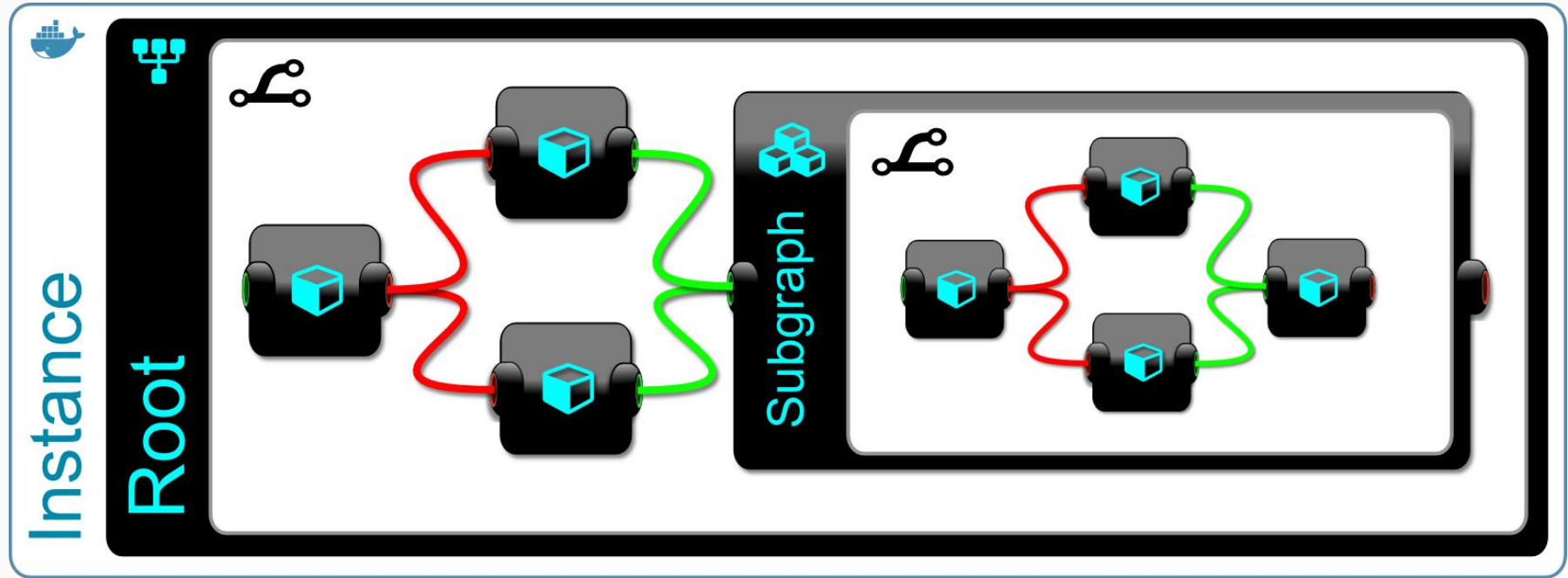


Subgraphing



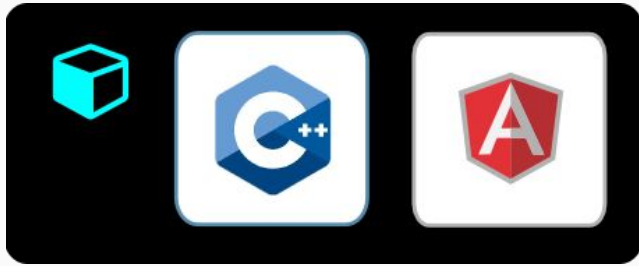
Any component can have a subgraph. When a component has a subgraph, it is called a parent component (the subgraph components are its children). Any non-connected ports can be exposed as parent component ports via port remapping.

Root Component

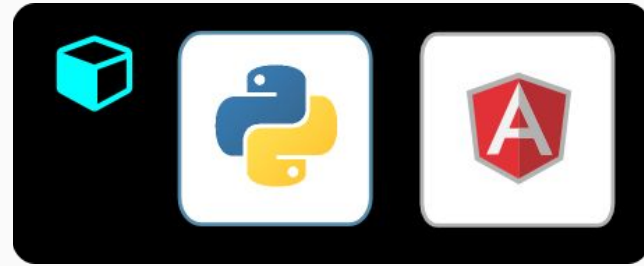


A root component has a subgraph, a startup script, and acts as an entry point for a StrategyLab instance.

Component Types

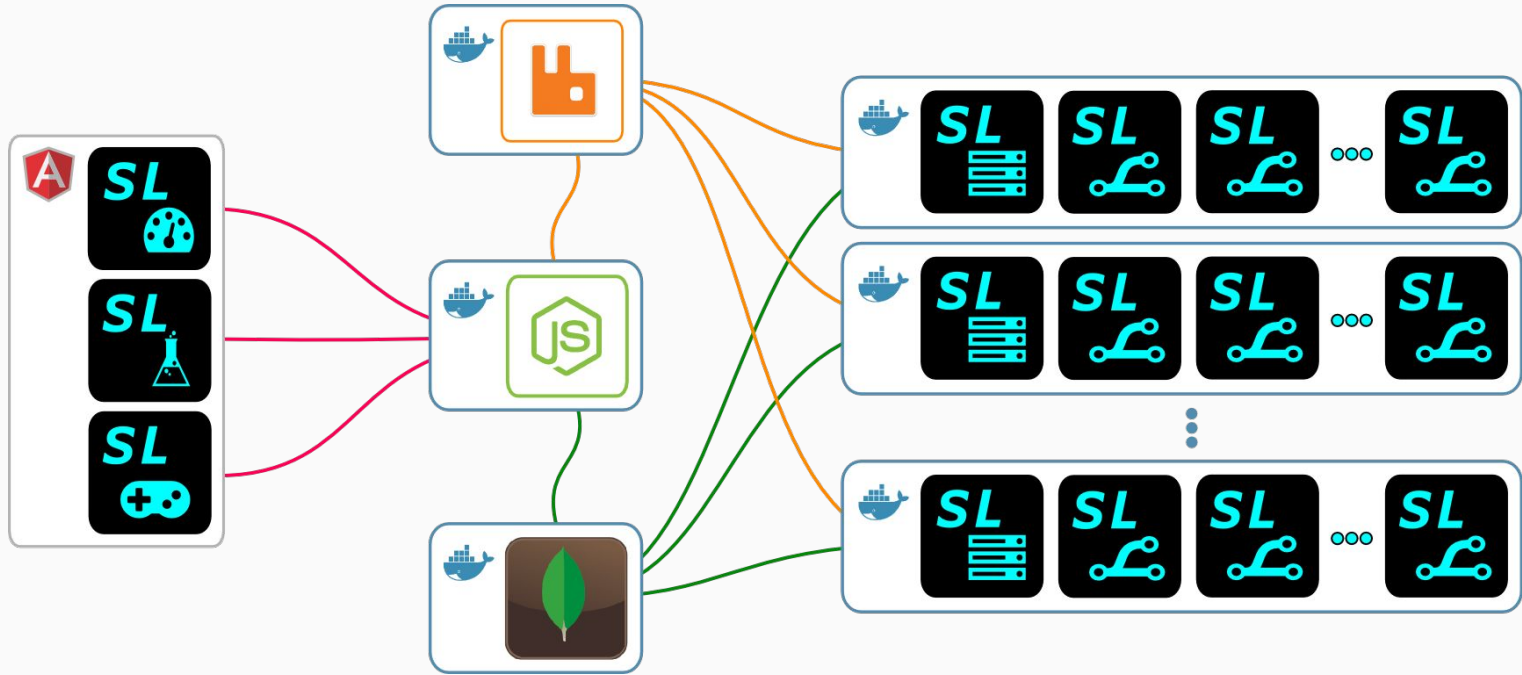


Server: C++
Client: Angular

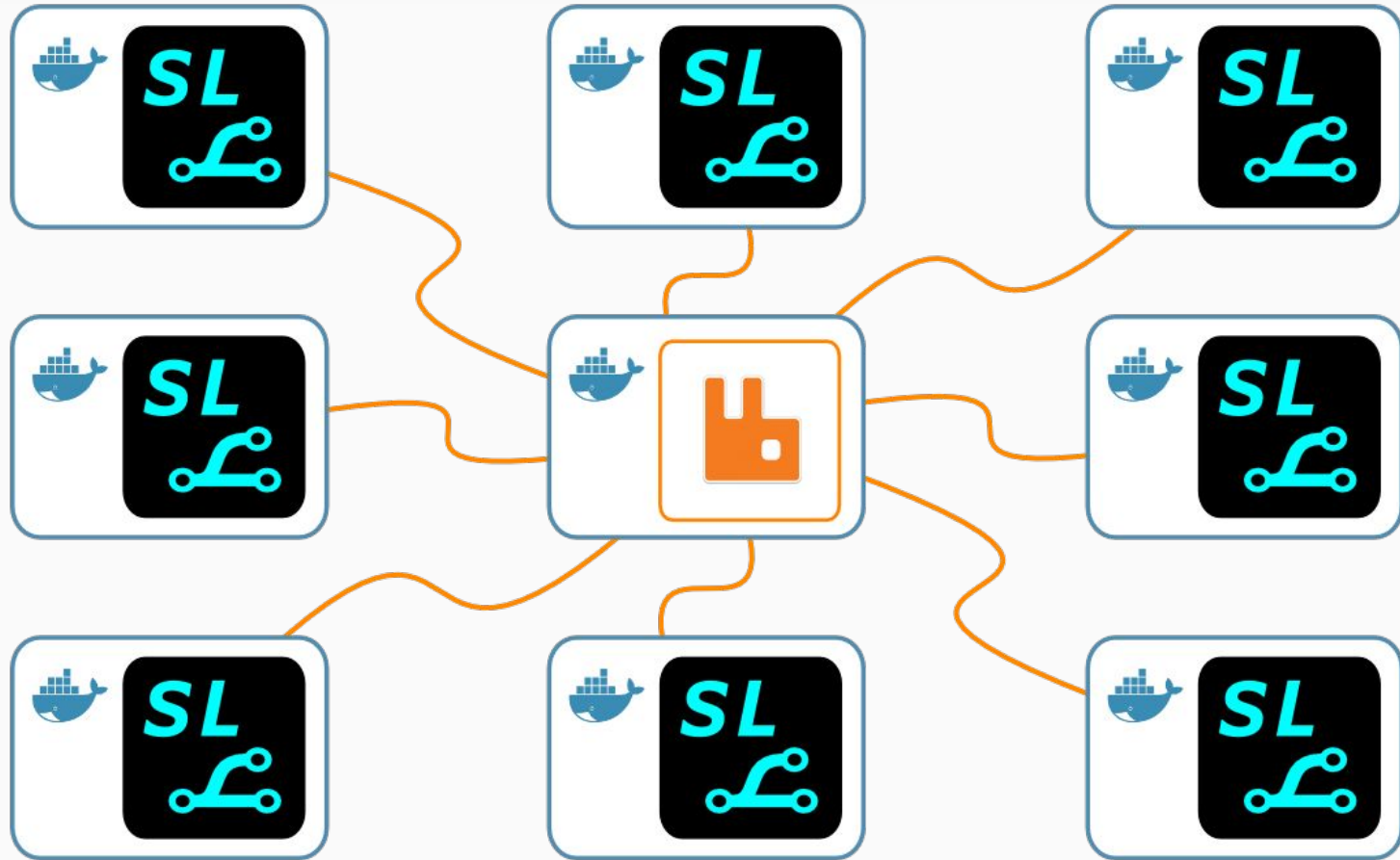


Server: Python
Client: Angular

Cloud Runtimes



Intergraph RMQ messaging:



Technologies



AngularJS



NodeJS



RabbitMQ



MongoDB



Docker



Runtime



Instance



Lab



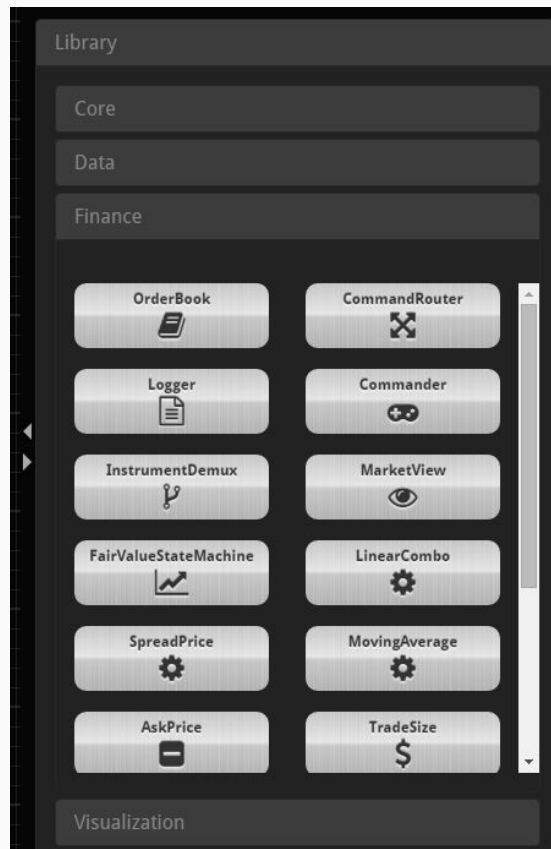
Visual



Controller

In Development:

Core
Data
Finance
Logic
Math
Messaging
Visualization

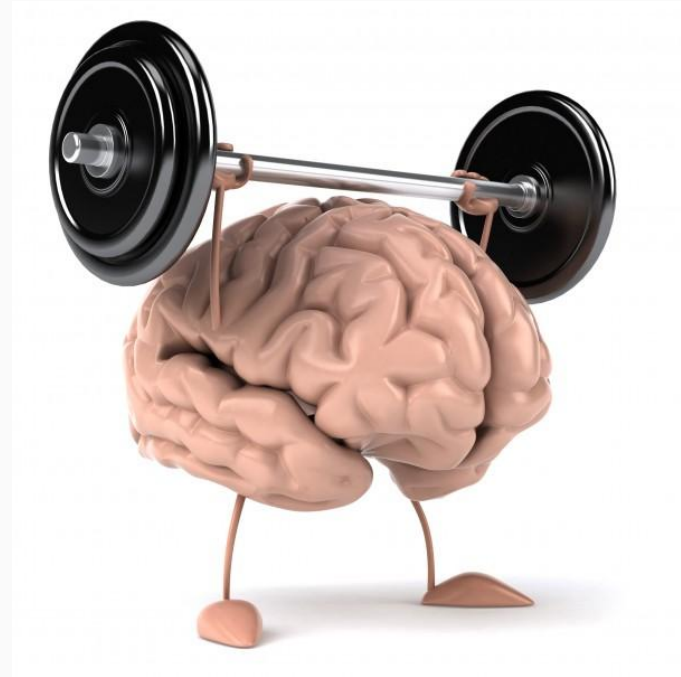


To Do:

Concurrency
Machine Learning
Optimization
Signal Processing
Utilities

Strengths

- Docker Scalability
- CPP Performance
- Python Rapid Development
- Browser Based
- Mobile to Multiple Monitors
- Physical Mobility
- Extremely Modular
- Auto Generating GUI



Weaknesses

- Security
- Module Depth/Diversity
- Social Platform (public, private and shared components/graphs)
- Source Control Integration and Versioning



Opportunities

Types:

- Complex Event Processing
- Machine Learning
- Grid Computing
- Big Data Processing

Industries:

- Finance
- Advertising
- Defense
- Big Data
- Social Media
- Science and Research



Big Data



Data collection

- Hardware
- Networking
- Infrastructure
- Datacenters



Data Processing

- Storage
- Database
- Technology Influence (In-memory, NoSQL, Hadoop, R, Map-reduce etc.)



Data Analysis

- Analytics
- Predictive
- Datawarehouses
- But also as a service, cloud, mobile etc.



Data Execution

- Services
- Integration
- Specialized Vars
- Large ISVs

\$50 billion market in 2017

Front End:

Basic System Licensing
Component and Module Licensing

Back End:

Dedicated Hardware Hosting
Shared Hardware Hosting
Cluster Time
StrategyLab Appliance

Consulting:

Domain Language Design
Component Development
Private Cloud Setup



Competition:

- Matlab + Simulink \$\$\$
- LabVIEW \$\$\$
- TIBCO Streambase \$\$\$\$
- NoFlo \$



Year 1 Budget \$1,615,000:

Personnel (Base):

- ❑ \$125,000: Web Developer
- ❑ \$150,000: C++ / Python Developer
- ❑ \$100,000: Dev Ops Engineer
- ❑ \$125,000: Module Developer
- ❑ \$180,000: Chief Architect
- ❑ \$180,000: Chief Operations Officer

Misc:

- ❑ \$120,000: Office Space
- ❑ \$50,000: Furniture
- ❑ \$215,000: Bonuses
- ❑ \$170,000: Health Care

Hardware, Networking and Hosting:

- ❑ \$50,000: Office Workstations & Networking
- ❑ \$50,000: Dev Server Environment
- ❑ \$100,000: Prod Server Environment

Revenue Model:

Monthly Recurring Revenue

- \$5,000: Base System
- \$1,000: Per Module Extension
- \$2,500: Dedicated Runtime
- \$10,000: Appliance
- \$1,000: Connectivity (MD, web, etc)

Client A (\$13,500 Monthly):

Hosted

- \$5,000: Base System
- \$5,000: 5 Module Extensions
- \$2,500: 1 Dedicated Runtime
- \$1,000: 1 Data source

Client B (\$27,500 Monthly):

Private Install

- \$5,000: Base System
- \$2,000: 2 Module Extensions
- \$20,000: 2 Appliances

With a \$135k monthly burn rate, we break even with 2 x Client B relationships and 5 x Client A relationships.