QuickFIX 101

Building a Sample Trading App

Mike Gatny

Vice President - Consulting Services

connamara systems, LLC

Introduction

Chicago FinTech Open Source Meetup: QuickFIX 101

About Mike Gatny

- Lead Engineer and VP Consulting Services at Connamara, since ‘06
- Contributor to the **QuickFIX/C++** and **OpenFAST** projects, and co-author of **QuickFIX/n**
- Co-author of **FixPresso**, Connamara’s proprietary, low-latency rewrite of **QuickFIX/j**, used by a major exchange

About Connamara Systems

- Software Engineering firm specializing in FinTech, est. 1997
- Experts in **Agile** software development
- Projects include: HFT systems, risk mgmt systems, exchanges, matching engines, front-end trading screens
- Clients include: hedge funds, CTAs, FCMs, exchanges, etc.
- Involved with **QuickFIX/C++** since the project began. Creators of **QuickFIX/n**, **agent_zmq**, **agent_fix**, **protobuf_spec**, and **fix_spec**
- Paid support for all flavors of QuickFIX

Introduction

Chicago FinTech Open Source Meetup: QuickFIX 101

QuickFIX comes in a variety of flavors:

- **QuickFIX**: *C++*, with wrappers for *python*, *ruby*, *.NET*
- **QuickFIX/j**: *Java* implementation - The best version for Java
- **QuickFIX/n**: *C#* implementation - The best version for .NET
- **QuickFIX/go**: *Go* implementation - Coming soon from Connamara

What kinds of things do people use QuickFIX for?

- Automated trading systems that need to send orders & receive fills.
- Risk management systems that need to receive drop-copies of fill reports in order to track firm-wide positions & P/L.
- Trading screens that need to display financial market data.
- Exchanges, brokers, banks, etc. that want to provide a FIX protocol API, often on top of their existing proprietary protocols.

What we will cover

Chicago FinTech Open Source Meetup: QuickFIX 101

How to organize a QuickFIX application

- E.g. for trading apps that need to receive market data, send orders, and receive execution reports.

How to add custom settings to the QuickFIX config file

- E.g. for adding a Password to every Logon message.

How to customize the QuickFIX DataDictionary file

How to create a sim environment for a FIX app by:

- Hooking it up to the *Executor* example app from the QuickFIX project
- Hooking it up to a simple, purpose-built FIX market data simulator

How we will cover it

Chicago FinTech Open Source Meetup: QuickFIX 101

Create a *Workspace* for running a trading *Strategy*

Create a simple momentum strategy in the Workspace

Hook it up to the *Executor* and a market data Simulator

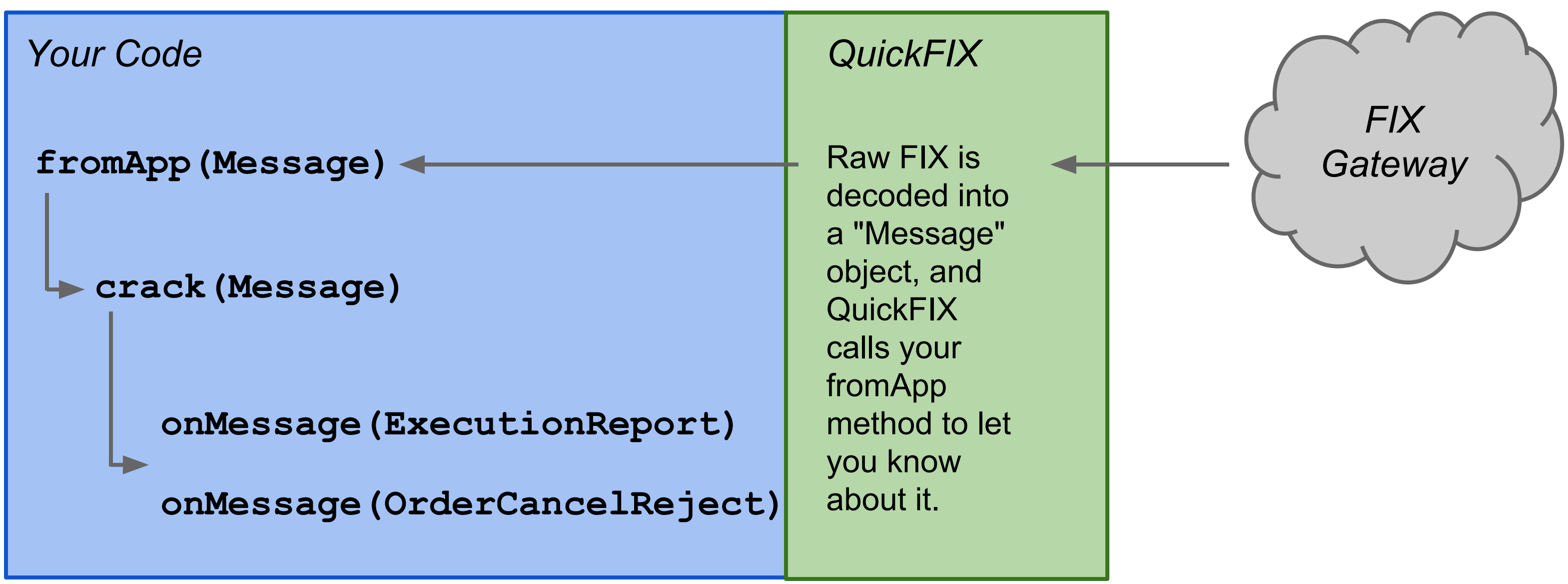
QuickFIX data flow

Chicago FinTech Open Source Meetup: QuickFIX 101

So, how does QuickFIX notify our application that an interesting event has occurred?

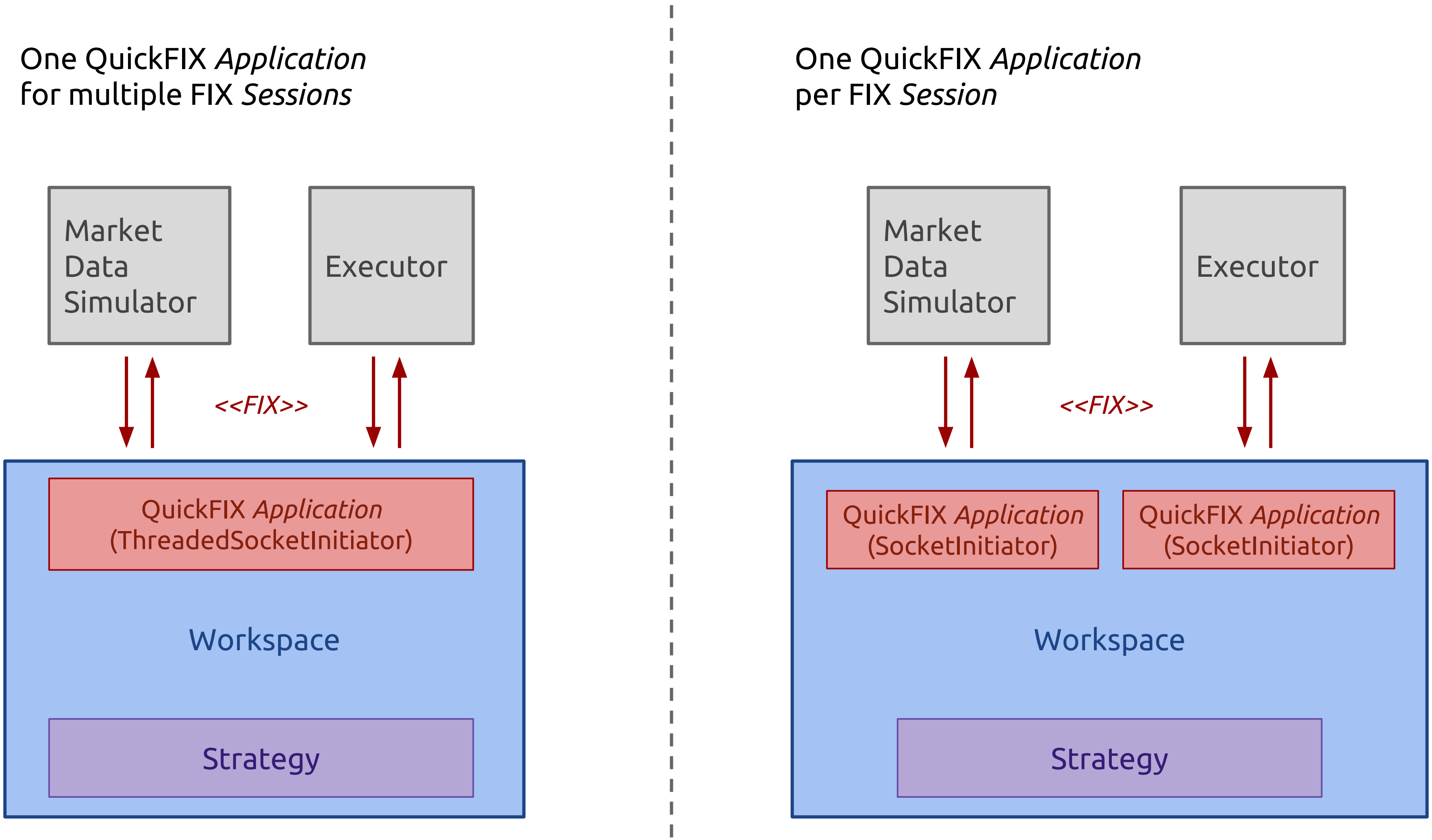
We implement *callbacks* that QuickFIX will invoke to deliver events to us.

It looks something like this:



App architecture

Chicago FinTech Open Source Meetup: QuickFIX 101



Basic app logic

Chicago FinTech Open Source Meetup: QuickFIX 101

Connect Strategy to Workspace

Send market data subscription

Listen for market data updates

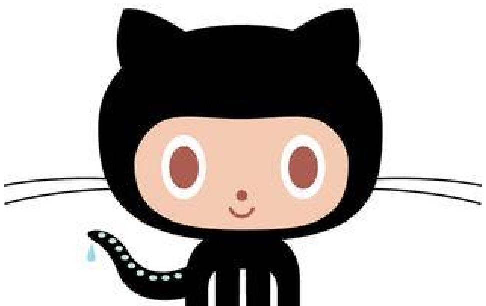
Listen for execution reports

When a TRADE msg is received, decide whether to send orders

When a FILL msg is received, update our position

Follow along

Chicago FinTech Open Source Meetup: QuickFIX 101



https://github.com/mgatny/quickfix_101

Where to get help

Start with the QuickFIX example apps!

Always check your QuickFIX ***messages.log*** and ***event.log***

Ask questions on the QuickFIX mailing list

See: http://quickfixengine.org/quickfix/doc/html/

Get a Connamara QuickFIX Support Bundle!