Low Latency Java in the Real World

LMAX Exchange and the Zing® JVM

Mark Price, Senior Developer, LMAX Exchange
Gil Tene, CTO & co-Founder, Azul Systems
Matt Schuetze, Product Manager, Azul Systems

About the Author: Gil Tene

- co-founder, CTO @Azul Systems
- Has been working on "think different" GC approaches since 2002
- A Long history building Virtual & Physical Machines, Operating Systems, Enterprise apps, etc…
- Depresses people by demonstrating how terribly wrong their latency measurements are…

* working on real-world trash compaction issues, circa 2004

About the Author: Mark Price

- Senior Developer at LMAX Exchange
- Performance enthusiast
- Touched pretty much everything at LMAX over a period of 8 years…
- Currently focusing on performance and monitoring
- The guy that everyone used to come to with their GC logs…

About me: Matt Schuetze

- Product Management Director at Azul
- Translate Voice of Customer into Zing and Zulu requirements and work items
- Sing the praises of Azul efforts through product launches
- Azul alternate on JCP exec committee, co lead Detroit JUG
- Stand on the shoulders of giants and admit it
- Riding the Java hype cycle roller coaster on a bicycle…

Background

- LMAX Exchange is a venue for low-latency FX trading
- World's first regulated MTF for transparent, order-driven FX trading
- 1st place in Tech Track 100 2014
- We aim to be the world's fastest retail FX exchange
- Open culture. A lot of open source

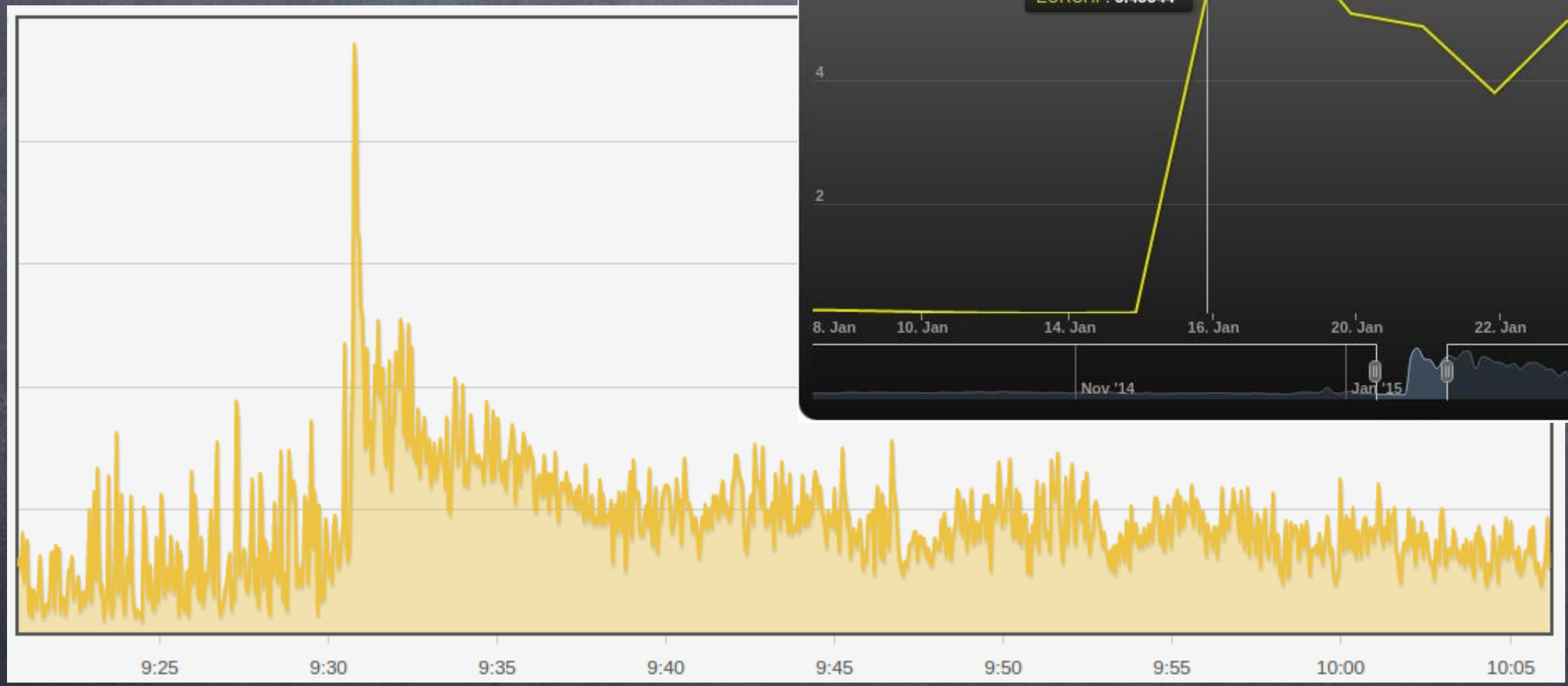
Zing: what is it good for?

- Not just fast. Always fast.
- Eliminates GC as a problem/concern
  - Addresses all forms of GC pauses, large & small
    - Low latency or Human-scale latency
    - Small or Large heaps, Low or High throughput
    - GC is simply a solved problem. Move on…
- Idiomatic Java code "just works"
  - Even when blips and pauses are not an option

LMAX EXCHANGE

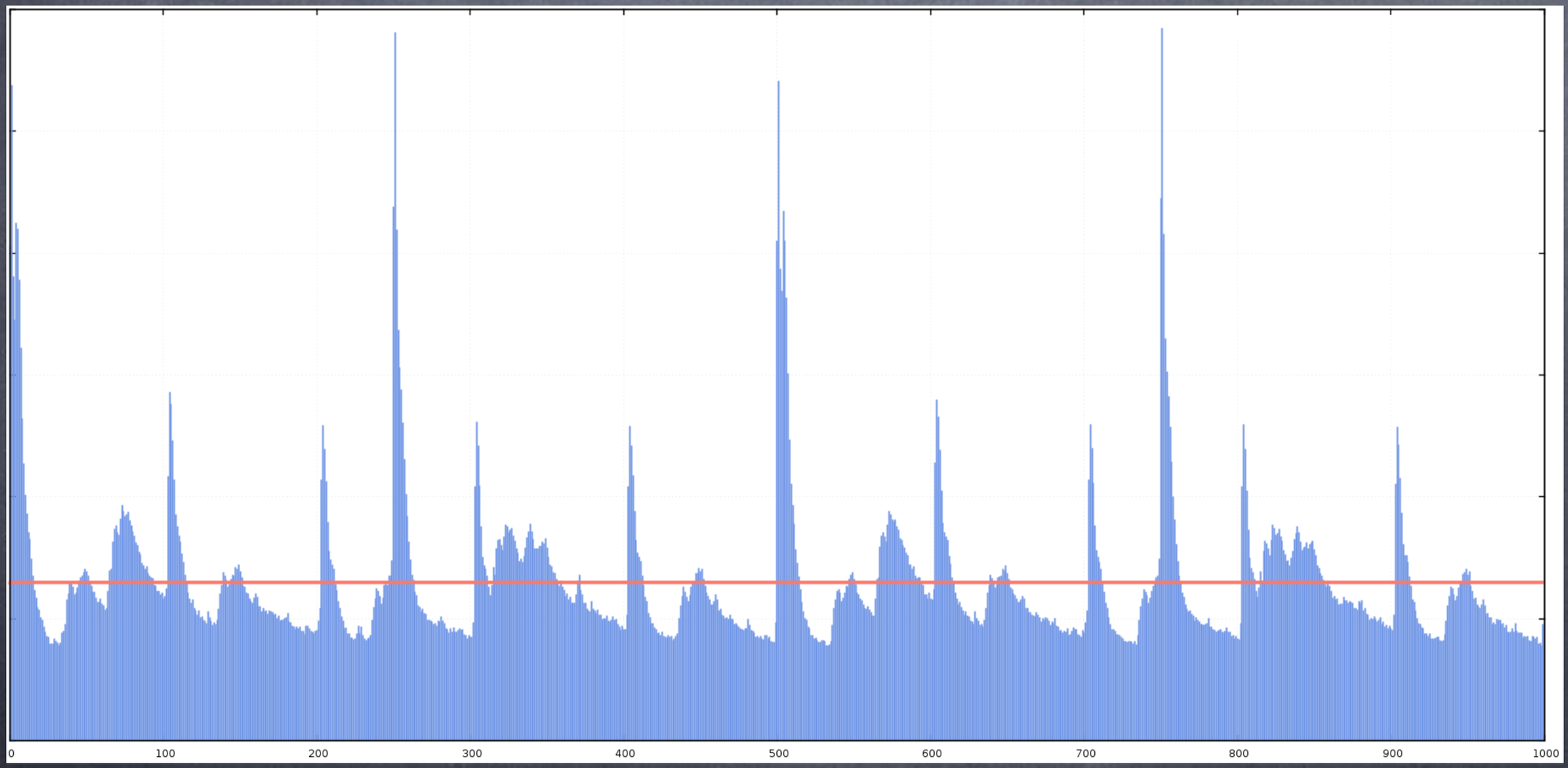
AZUL SYSTEMS

Stay Responsive

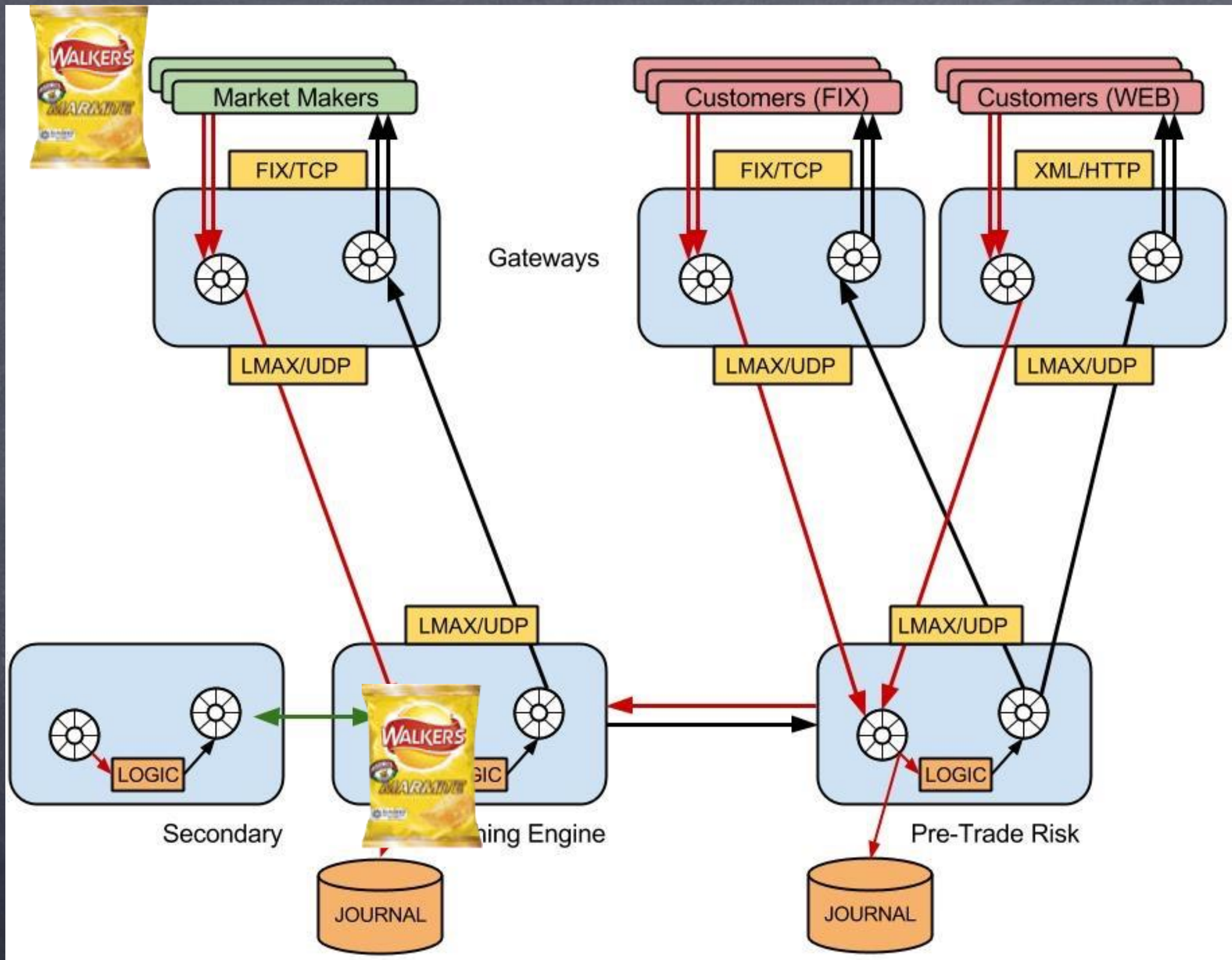


Even when traffic patterns change

Real world traffic patterns

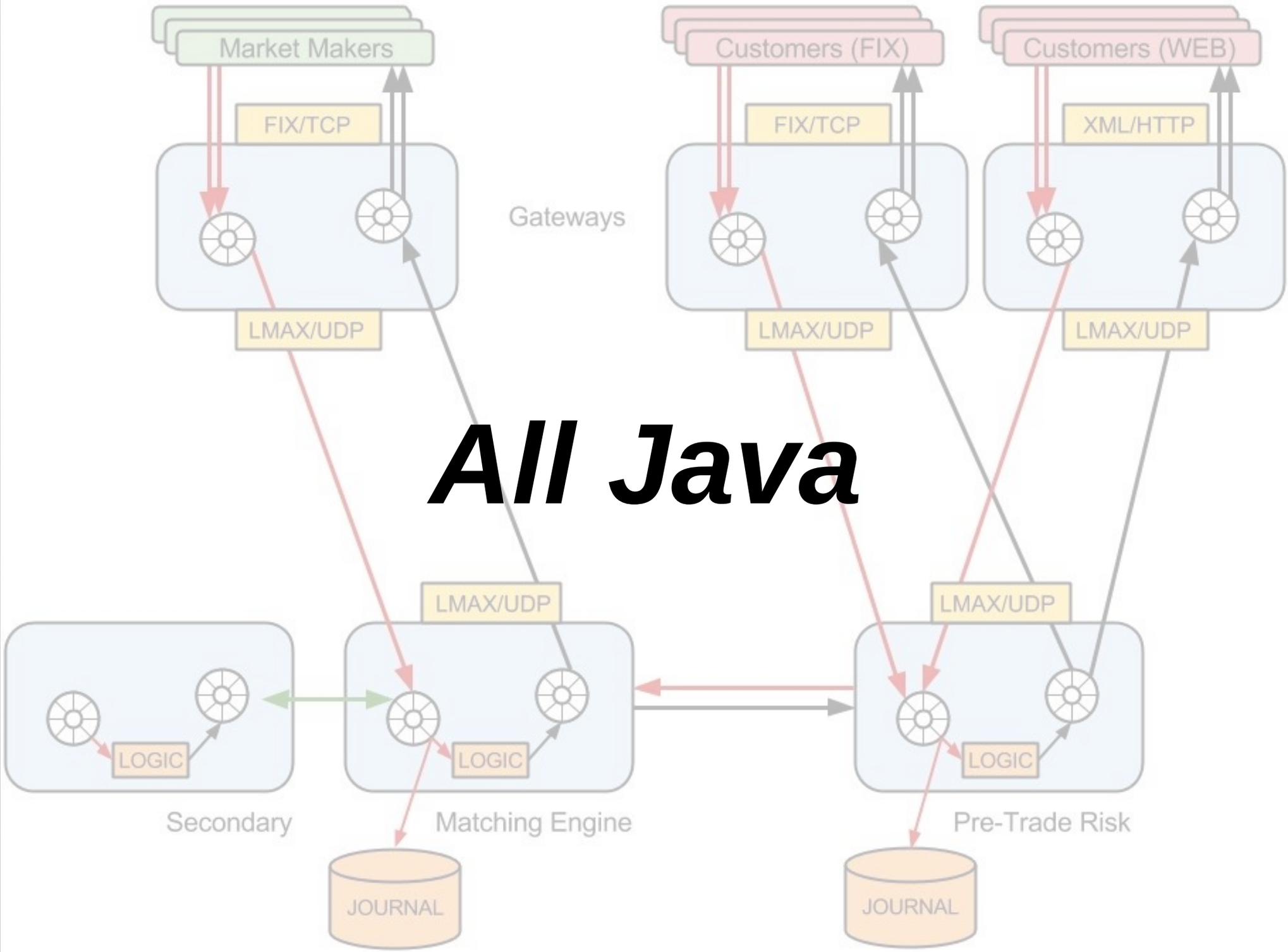


Not constant. Not random either.



Java for low latency

- Seriously?
- Yes...
  - Time to Stability
  - Time to Market
  - Time to Performance
- Overall productivity and delivery benefits trump the various downsides



All Java

The good, the bad, and the ugly

- Java is fast. Java is productive.
  - Good developers will produce good, fast code
  - Bad developers will produce bad, slow code
  - More Java devs than C/C++ devs (good & bad)
- But JVMs are [normally] not consistently fast
  - GC pauses & Deoptimization & Deflation, oh my.
- Low latency "Java" usually:
  - Written in the Java syntax. Avoids idiomatic Java.
  - Avoid using anyone else's code...

Zing's impact on low latency Java

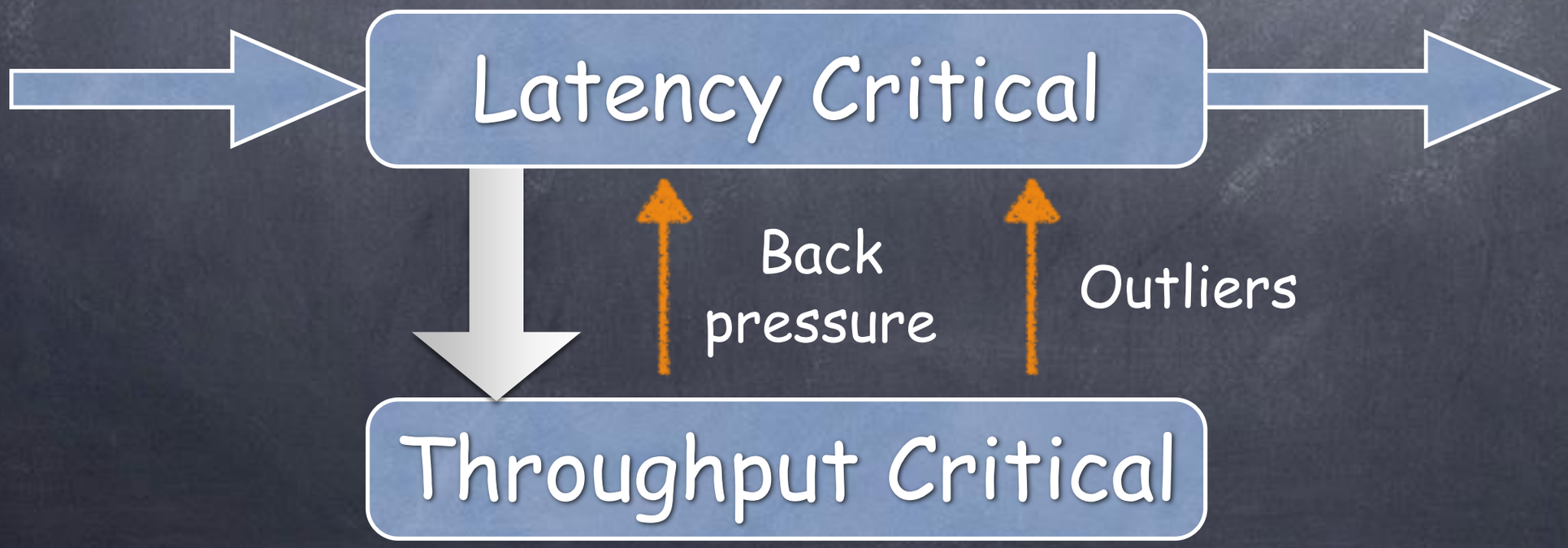
- Zing is a JVM that values consistent speed
  - GC (and JVM) "noise" reduced to below OS "noise"
- Keeps the good
- Removes the bad and the ugly
- Eliminates GC-related engineering efforts
  - All the associated bad and ugly choices
  - Idiomatic Java, not just "Java"

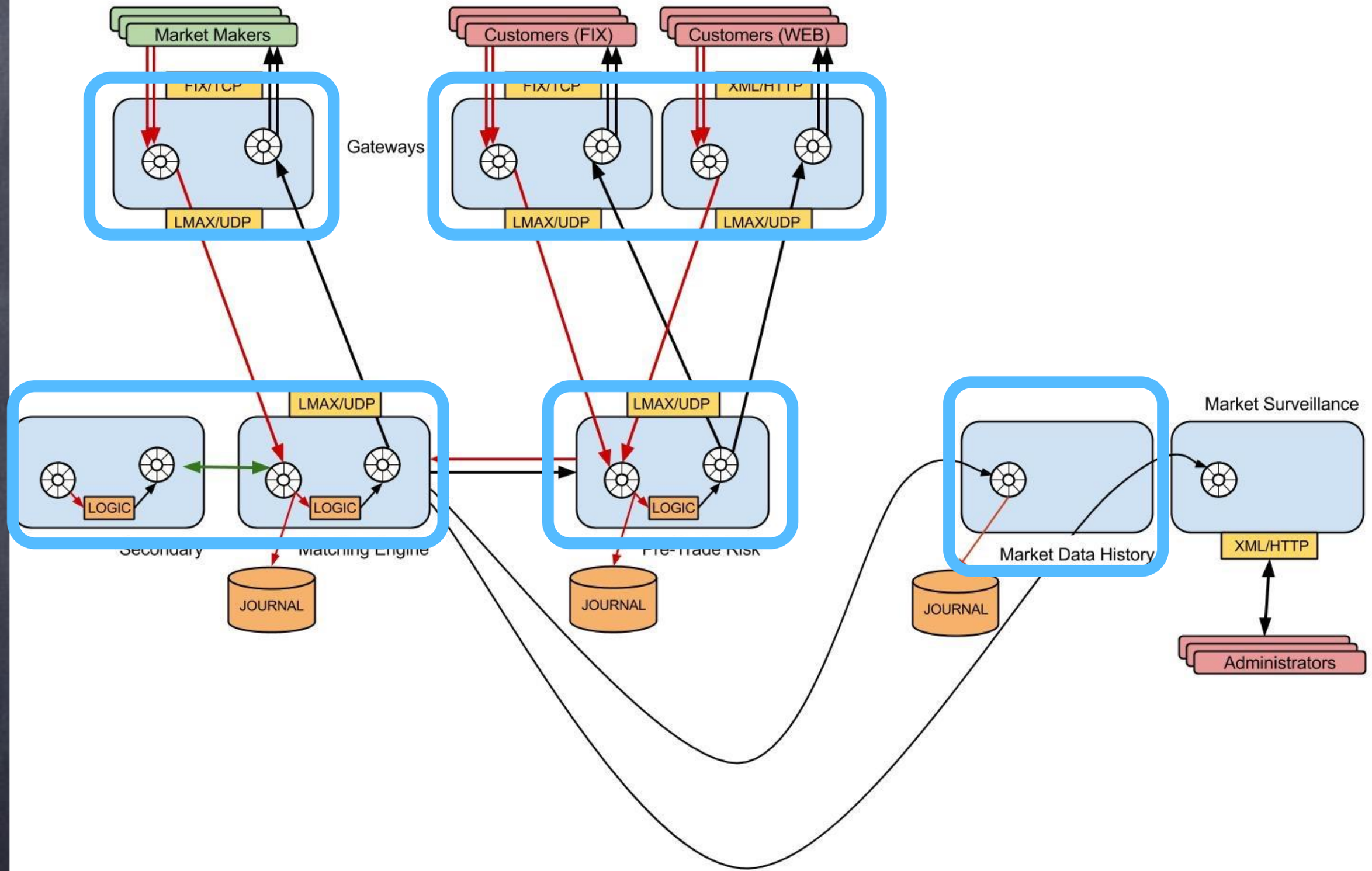
LMAX EXCHANGE

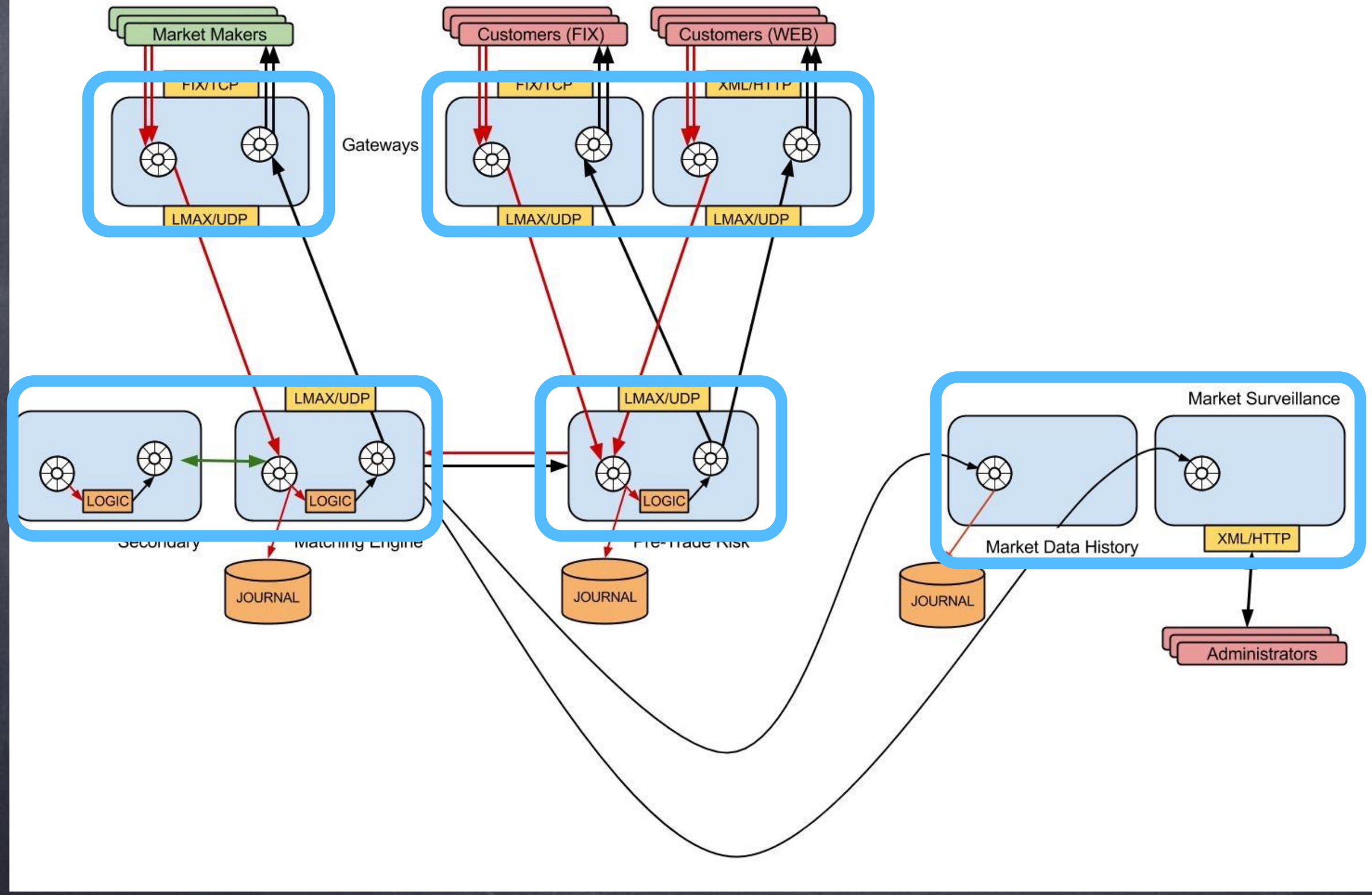
AZUL SYSTEMS

Zing at LMAX Exchange

- Started 2 years ago
- Incremented from most critical to less critical
- Next cover all latency-sensitive system
- Also cover throughput critical systems







Development at LMAX Exchange

- Heavy focus on TDD and CI
- 9000+ acceptance tests
- Performance is treated as just another part of CI
- Regression in performance same as a functional failure
- Confidence to make large changes

Zing at LMAX Exchange: Benefits

- Improved latency behaviors
- Reduced engineering effort
  - No more battling GC in code
  - Performance tuning cycles shorter
- Idiomatic Java is a big deal
  - can avoid converting to "special practices" java
  - can "ease up" on no-allocation coding styles…
  - keep using/leveraging 3rd party code
  - more productive overall…

LMAX EXCHANGE

AZUL SYSTEMS

Lessons Learned

- Attacking critical parts first:
  - may delay things…
  - critical parts have had engineering effort applied
  - smaller initial gains
  - longer time to gain confidence
- With hindsight:
  - Could have started from "outside in" instead
  - Larger initial gains, shorter time-to-confidence
  - May have gotten to wider deployment faster

Lessons Learned: GC is not the only problem

- GC does tend to dominate outliers (pre-Zing)
  - it is natural to assume all big bad stuff is GC
  - with GC outliers gone, remaining problems surface
- Some harder things:
  - page cache (kernel) lock contention
  - power management tuning in BIOS & OS
  - mapped file access faults & safepoints

Lessons Learned: Proactively override Linux defaults

- Out-of-the box Linux has some bad defaults
  - Bad for things that care about latency…
- E.g. all these should be changed upfront:
  - Page cache: vm.min_free_kbytes [needs >1GB]
  - Transparent Huge Pages (THP) [should be turned off]
  - Swappiness [should be set to 0]
  - zone_reclaim_info [should be set to 0]
- Each can cause multi-100s-of-msec outliers

LMAX EXCHANGE

AZUL SYSTEMS

Lessons Learned: Measure, Measure, Measure

- We were already measuring & collecting a lot
- But jitter/hiccup/outlier measurement is tough
  - Especially across multiple hops
- Detailed latency distribution measurements
  - Helped triage outlier issues and focus efforts
  - Full percentile spectrum histograms
  - Record system-level "hiccups" everywhere

Real World Results

- Zing helped LMAX tame GC-related latency outliers
  - Highly-engineered system: 4ms every 30 seconds down to 1ms every 2 hours
  - Less well-tuned system: 50ms every 30 seconds down to 3ms every 15 mins
- No more unexpected/unwanted old-gen pauses caused by external behaviour
  - CMS STW intra-day, generally ~500ms, gone
  - Removed source of backpressure on latency critical path.
  - Pre-Azul these would occur less predictably, but multiple times a week.

Summary

- Java is both viable and profitable for low latency
- With "regular JVMs" you have to jump through some hoops to get there
- Zing helps low latency in Java be as easy as low latency in other languages
- Once you stop dealing with JVM-related issues, you can get on with the interesting stuff

LMAX EXCHANGE

AZUL SYSTEMS

Azul Systems:
azul.com
More on Zing:
azul.com/zing
Twitter: @azulsystems

Speaker Contact:
Matt Schuetze
Product Manager, Azul Systems
@schuetzematt