



# JavaOne™

[java.sun.com/javaone](http://java.sun.com/javaone)

## LINKEDIN COMMUNICATION ARCHITECTURE

Ruslan Belkin, Sean Dawson

TS-5234



Learn how we at LinkedIn built and evolved scalable communication platform for the world's largest professional network

A large, light blue arrow pointing to the right, positioned behind the word "GOAL".

GOAL

# Agenda

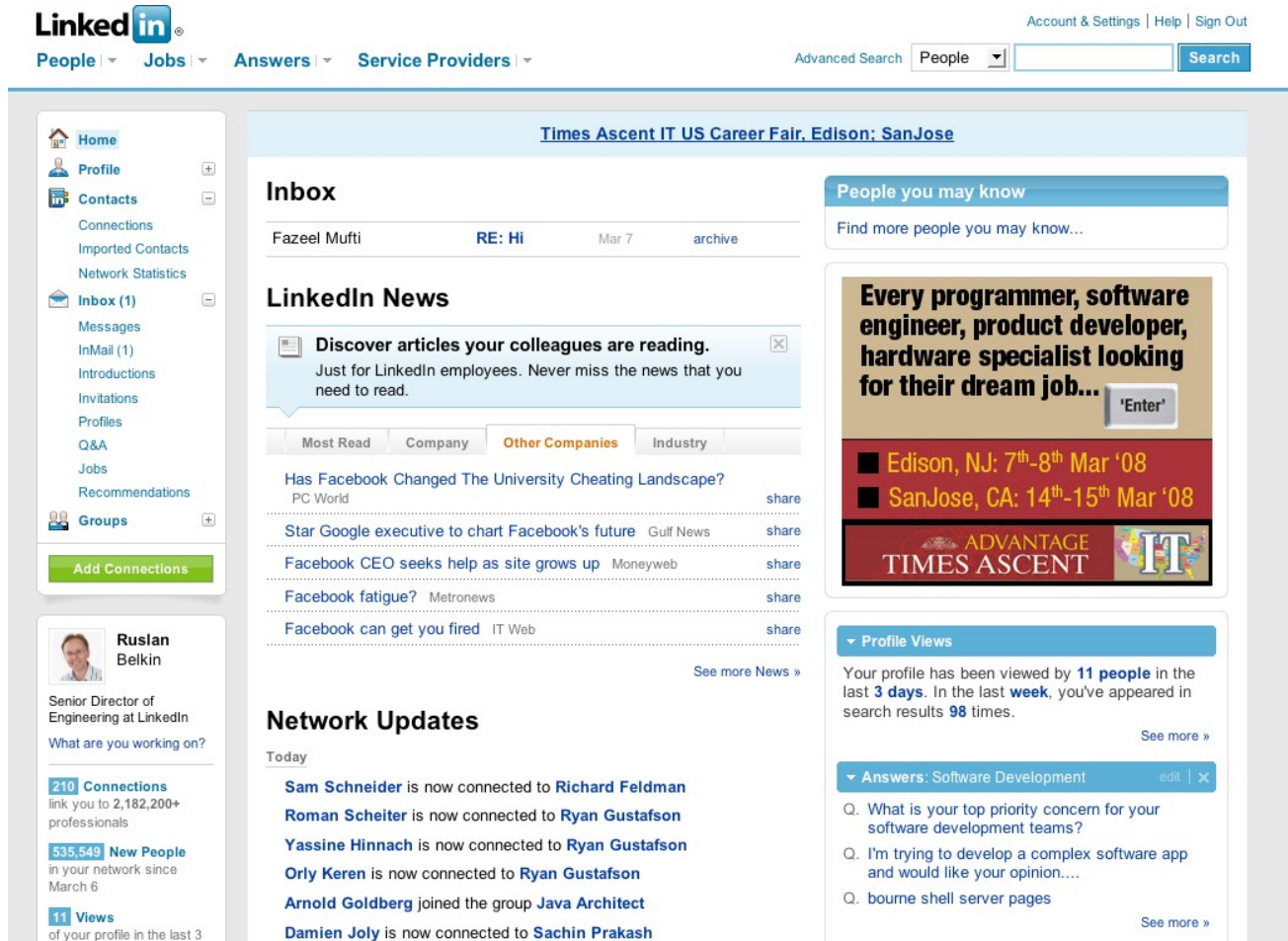
- Why are we doing this talk
- LinkedIn Communication Platform at a glance
  - Evolution of LinkedIn Communication System
  - Evolution of the Network Updates System
- Scaling the system: from 0 to 22M members
- Q&A

# Why are we doing this talk?

- Share our experience in building the world-largest professional network in Java™
- Describe the evolution of the communication platform
- Share lessons we learned so you could benefit from our successes, mistakes and experience

# LinkedIn Communication Platform

## Quick Tour

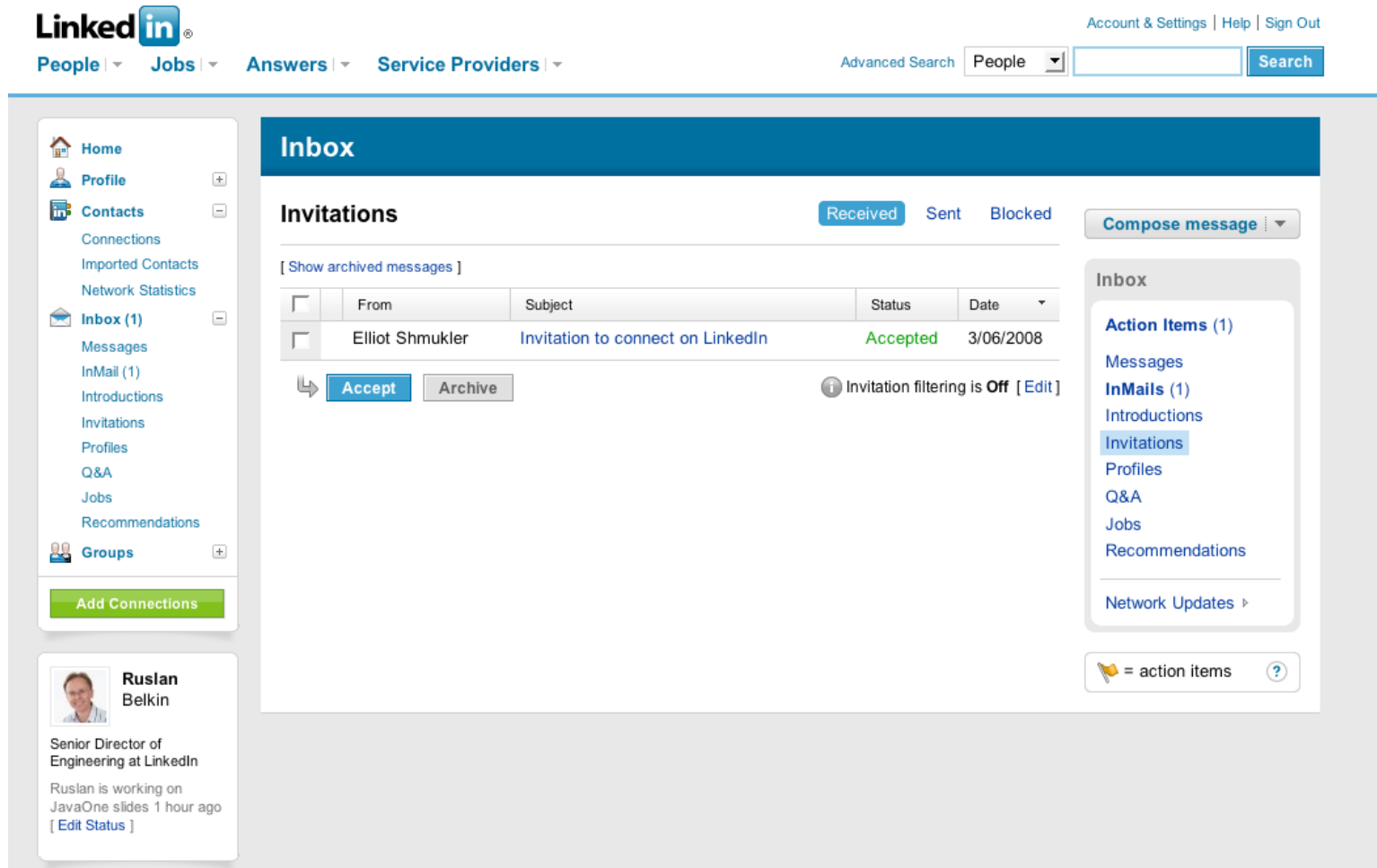


The screenshot shows the LinkedIn homepage with the following elements:

- Navigation Bar:** Includes the LinkedIn logo, links for People, Jobs, Answers, and Service Providers, and a search bar with a dropdown menu set to 'People'.
- Sidebar:** Contains links for Home, Profile, Contacts, Connections, Imported Contacts, Network Statistics, Inbox (1), Messages, InMail (1), Introductions, Invitations, Profiles, Q&A, Jobs, Recommendations, and Groups. There is also a 'Add Connections' button.
- Main Content Area:**
  - Inbox:** Shows an email from Fazeel Mufti with the subject 'RE: Hi' dated Mar 7.
  - LinkedIn News:** A section titled 'Discover articles your colleagues are reading.' with a list of news items:
    - Has Facebook Changed The University Cheating Landscape? (PC World)
    - Star Google executive to chart Facebook's future (Gulf News)
    - Facebook CEO seeks help as site grows up (Moneyweb)
    - Facebook fatigue? (Metronews)
    - Facebook can get you fired (IT Web)
  - Network Updates:** A section titled 'Today' showing recent connections:
    - Sam Schneider is now connected to Richard Feldman
    - Roman Scheiter is now connected to Ryan Gustafson
    - Yassine Hinnach is now connected to Ryan Gustafson
    - Orly Keren is now connected to Ryan Gustafson
    - Arnold Goldberg joined the group Java Architect
    - Damien Joly is now connected to Sachin Prakash
  - People you may know:** A section with a search bar and a list of suggested connections.
  - Every programmer, software engineer, product developer, hardware specialist looking for their dream job...:** A promotional banner for Times Ascent IT US Career Fair, Edison: San Jose, with dates for Edison, NJ (7th-8th Mar '08) and San Jose, CA (14th-15th Mar '08).
  - Profile Views:** A section showing that the user's profile has been viewed by 11 people in the last 3 days.
  - Answers:** A section titled 'Software Development' with a list of questions and answers.

# LinkedIn Communication Platform

## Quick Tour



The screenshot shows the LinkedIn communication platform interface. At the top, there's a navigation bar with the LinkedIn logo, links for "People", "Jobs", "Answers", and "Service Providers", and a search bar. Below the navigation bar, the main content area is divided into three sections: a left sidebar, a central inbox, and a right sidebar.

**Left Sidebar:** Contains navigation links for "Home", "Profile", "Contacts", "Connections", "Imported Contacts", "Network Statistics", "Inbox (1)", "Messages", "InMail (1)", "Introductions", "Invitations", "Profiles", "Q&A", "Jobs", "Recommendations", and "Groups". There is also a green button labeled "Add Connections".

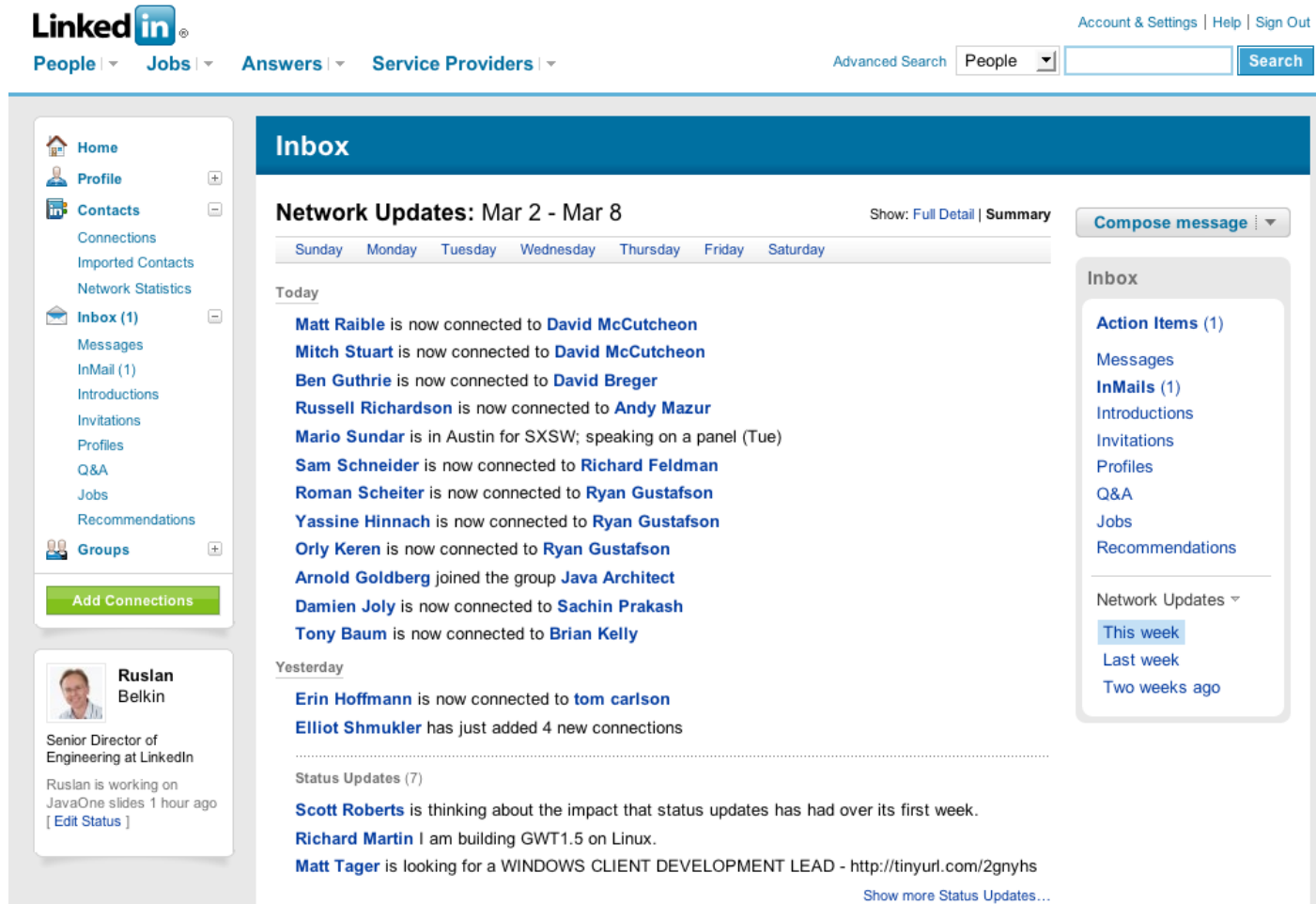
**Central Inbox:** The main section is titled "Inbox" and shows a list of "Invitations". The list has columns for "From", "Subject", "Status", and "Date". A single invitation is visible from "Elliot Shmukler" with the subject "Invitation to connect on LinkedIn" and status "Accepted". Below the invitation, there are "Accept" and "Archive" buttons. A message "Invitation filtering is Off [Edit]" is displayed.

**Right Sidebar:** Contains a section titled "Inbox" with a list of "Action Items (1)". The list includes "Messages", "InMails (1)", "Introductions", "Invitations" (which is highlighted), "Profiles", "Q&A", "Jobs", "Recommendations", and "Network Updates".

**Bottom Left:** A profile card for "Ruslan Belkin", Senior Director of Engineering at LinkedIn. It includes a photo, name, title, and a status update: "Ruslan is working on JavaOne slides 1 hour ago [Edit Status]".

# LinkedIn Communication Platform

## Quick Tour



The screenshot shows the LinkedIn homepage. At the top, there's a navigation bar with the LinkedIn logo, links for People, Jobs, Answers, and Service Providers, and a search bar. Below the navigation bar, the main content area is divided into three sections: a left sidebar, a central inbox, and a right sidebar.

**Left Sidebar:** Contains links to Home, Profile, Contacts, Connections, Imported Contacts, Network Statistics, Inbox (1), Messages, InMail (1), Introductions, Invitations, Profiles, Q&A, Jobs, Recommendations, and Groups. There is also a green button labeled "Add Connections".

**Central Inbox:** Titled "Inbox", it shows "Network Updates: Mar 2 - Mar 8". Below this, there's a tabbed interface for "Today" and "Yesterday". The "Today" tab lists several network updates, including connections between Matt Raible, David McCutcheon, Mitch Stuart, Ben Guthrie, David Breger, Russell Richardson, Andy Mazur, Mario Sundar, Richard Feldman, Sam Schneider, Roman Scheiter, Ryan Gustafson, Yassine Hinnach, Orly Keren, Ryan Gustafson, Arnold Goldberg, Java Architect, Damien Joly, Sachin Prakash, and Tony Baum. The "Yesterday" tab shows updates from Erin Hoffmann and Elliot Shmukler.

**Right Sidebar:** Contains a "Compose message" button, a list of "Action Items (1)", and a "Network Updates" section with links for "This week", "Last week", and "Two weeks ago".

**Bottom Left:** A profile card for Ruslan Belkin, Senior Director of Engineering at LinkedIn, with a link to "Edit Status".

# LinkedIn Communication Platform

## The Numbers

- 22M members
- 130M connections
- 2M email messages per day
- 250K invitations per day



# LinkedIn Communication Platform

## The Setup

- Sun™ x86 platform and Sparc production hardware running Solaris™ Operating System
- 100% Java programming language
- Tomcat and Jetty as application servers
- Oracle and MySQL as DBs
- ActiveMQ for JMS
- Lucene as a foundation for search
- Spring as a glue
- Mac for development

# LinkedIn Communication Platform

## ➤ The Communication Service

- Permanent message storage
- InBox messages
- Emails
- Batching, delayed delivery
- Bounce, cancellation
- Actionable content
- Rich email content

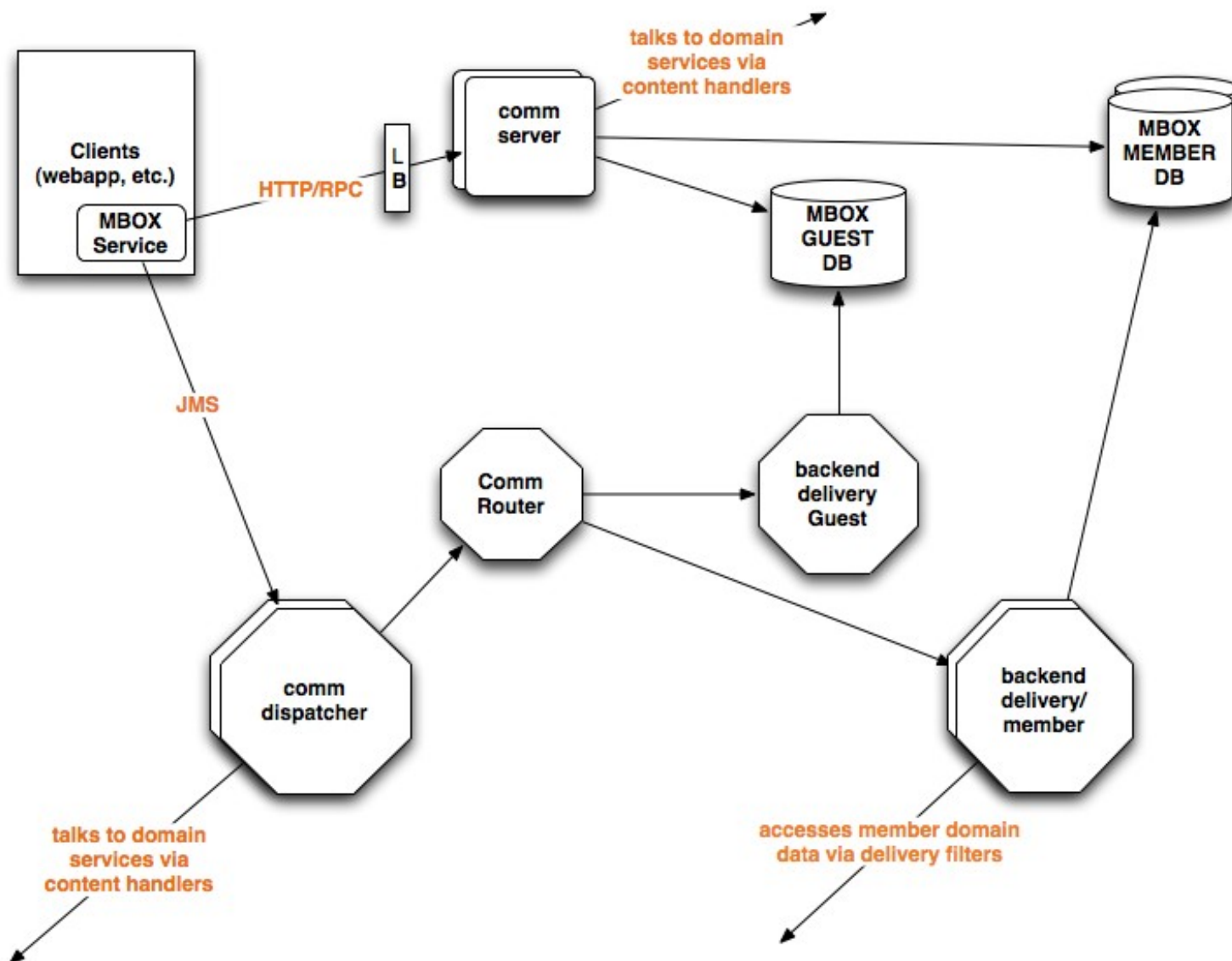
## ➤ The network updates service

- Short-lived notifications (events)
- Distribution across various affiliations and groups
- Time decay
- Events grouping and prioritization

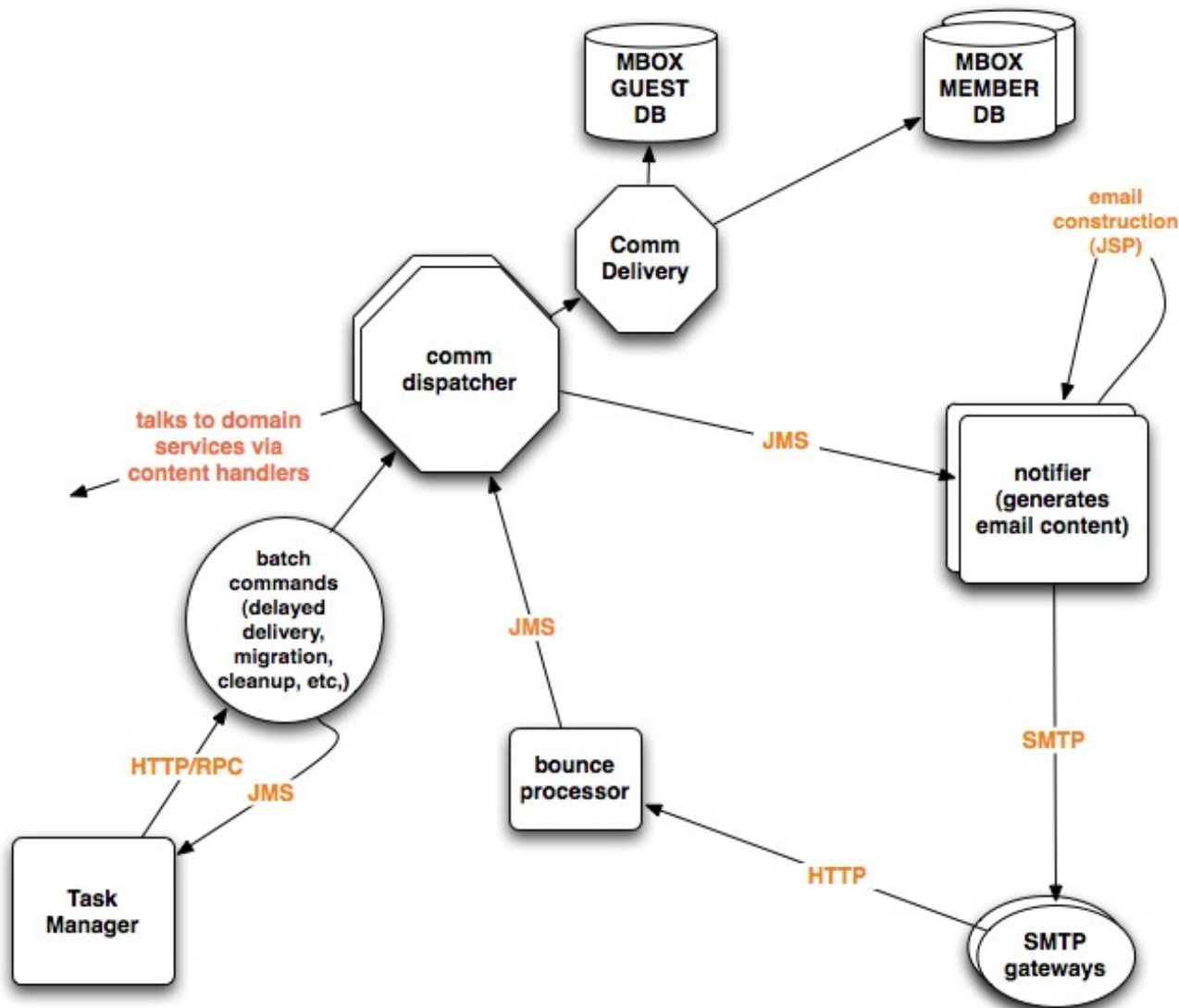
# The Communication Service

- How is it different:
  - Workflow oriented
  - Messages reference other objects in the system
  - Incorporates email delivery
  - Batching of messages
  - Message cancellation
  - Delayed delivery, customer service review queues, abuse controls
  - Supports reminders and bounce notifications to users
- Has undergone continuous improvements throughout life of LinkedIn

## Message Creation



# Message Delivery



# The Communication Service

## ➤ Message Creation

- Clients post messages via asynchronous Java Communications API using JMS
- Messages then are routed via routing service to the appropriate mailbox or directly for email processing
- Multiple member or guest databases are supported

## ➤ Message Delivery

- Message delivery is triggered by clients or by scheduled processes
- Delivery actions are asynchronous
- Messages can be batched for delivery into a single email message
- Message content is processed through the JavaServer Page™ (JSP™) technology for pretty formatting
- The scheduler can take into account the time, delivery preferences, system load
- Bounced messages are processed and redelivered if needed
- Reminder system works the same way as message delivery system

# The Communication Service

- SOA architecture
- Wireable components build around LinkedIn Spring extensions
- Spring HTTP-RPC
- Heavy use of JMS and asynchronous flows

# The Communication Service

## Failure Recovery

- Possible failures:
  - Messages can bounce
  - Messages can get lost:
    - Database problems
    - Bugs in the code
    - Bugs in the content processing of emails
    - Various services may become unavailable
- Avoiding the downtime



# The Communication Service

## How do we scale it?

- Functional partitioning:
  - sent, received, archived, etc.
- Class partitioning:
  - Member mailboxes, guest mailboxes, corporate mailboxes
- Range partitioning:
  - Member ID range
  - Email lexicographical range
- Asynchronous flows

# Network Updates Service


- What is your network up to?
- The goal is to have a flexible service for distributing many types of *short-lived* updates
- Availability across a number of clients (web apps, RSS, API, LinkedIn Mobile, third-party...)


# Network Updates Service

## Motivation

- Homepage circa 2007
- Poor UI
  - Cluttered
  - Where does new content go?
- Poor Backend Integration
  - Many different service calls
  - Takes a long time to gather all of the data


### Network updates since Sep 18




**Questions from your network:**


Asked by your connections:
   
[Colleen Cole: What are the best colocation facilities in the Bay Area?](#)

Asked by friends of friends:
   
[What's a good advertising company which specializes in online... ?](#)
  
[Where do most technology startups get their angel funding?](#)
  
[Do minimum wage laws apply to all companies?](#)
  
[View more questions »](#)

**Profile updates:**


- 
[Samuel Chung](#) has new profile information. [View Profile.](#)
  
[See all...](#)

[Chris Yee](#) is now a connection
   
[Billy Bob Thorton](#) is now a connection

[Chris Yee](#) has just added 1 connection
   
[Billy Bob Thorton](#) has just added 1 connection
   
[See all...](#)

**6,017 new people** in your network since September 18

# Network Updates Service

## Motivation

- Homepage circa 2008
- Clean UI
  - Eliminates contention for homepage real estate
- Clean Backend
  - Single call to fetch updates
  - Consistent update format

## Network Updates

### Today

**Kathryn Barrett** is now connected to **Barry Campbell**

**Surya Yalamanchili** is now connected to **Richard Chen**

### Yesterday

**Neil Bocalan** joined the group **San Francisco Giants Fan**

**Grig Gheorghiu** has just added 4 new connections

**Matt Raible** is now connected to **Dave Gullo**

**Adam Nash** joined the group **MoMIT - MIT's Club for Mobile, Media, and Internet Technology**

**Brad Lauster** is now connected to **Scotty Logan**

### Status Updates (5)

**Reid Hoffman** juggling madly at the office, as usual.

**Chris Saccheri** is back in the saddle again.

**David Sanford** is thinking about the next big thing

[Show more Status Updates...](#)

**David Sanford** has an updated profile (Headline, Expertise)

**Surya Yalamanchili** is now connected to **Abe Assad**

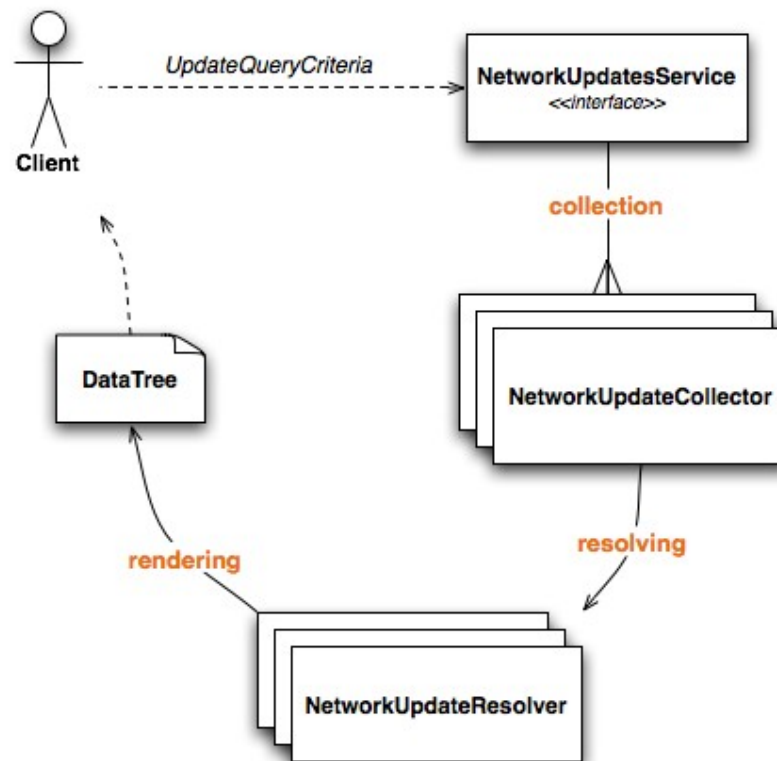
# Network Updates Service

## Iteration 1

- Move existing homepage logic into a remote service, refactor homepage to use the new service
- Advantages
  - No user impact while API is being finalized
  - Improve performance by fetching updates in parallel
  - Reduce complexity of the web app
  - Updates become easily accessible to other clients

# Network Updates Service

## Iteration 1 - API



# Network Updates Service

## Iteration 1 - API

- Pull-based architecture
- Collectors
  - Responsible for gathering data
  - Parallel collection to improve performance
- Resolvers
  - Fetch state, batch lookup queries, etc...
  - Use EHCache to cache global data (e.g., member info)
- Rendering
  - Transform each object into its XML representation

# Network Updates Service

## Iteration 1 - Example

```
UpdateQueryCriteria query =  
    UpdateQueryCriteria.createDefaultCriteria()  
        .setMemberID(2)  
        .setRequestedTypes(NetworkUpdateType.CONNECTION)  
        .setCutoffDate(...)  
        .setMaxNumberOfUpdates(10);
```

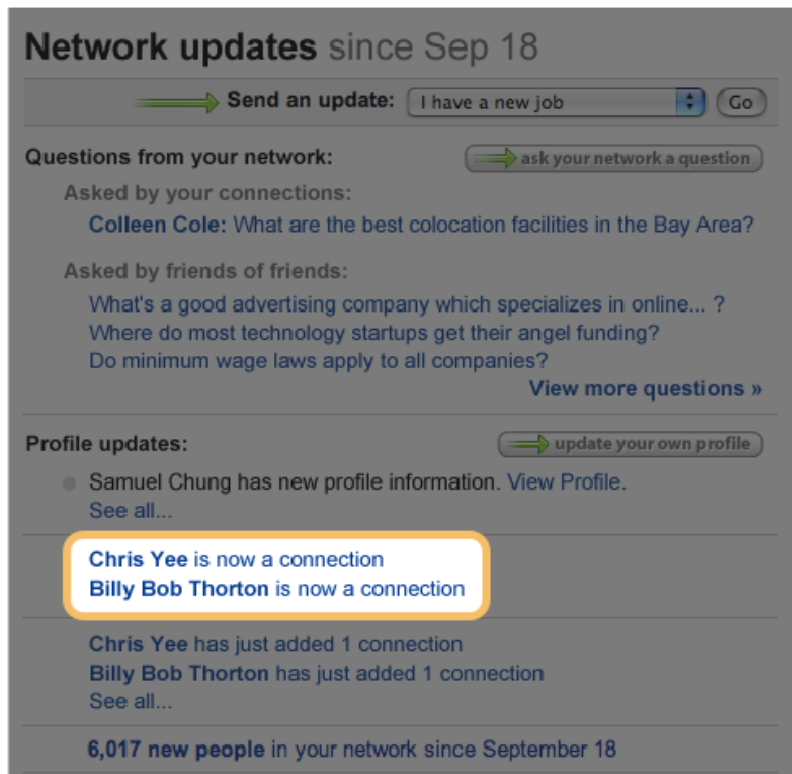
```
MyNetworkUpdatesService service =  
    getService(MyNetworkUpdatesService.class);
```

```
DataTree update =  
    service.getNetworkUpdatesSummary(query);
```



# Network Updates Service

## Iteration 1 - Example



```
<updates>
  <NCON>
    <connection>
      <id>2</id>

      <firstName>Chris</firstNa
me>

      <lastName>Yee</lastName>
    </connection>
  </NCON>
</updates>
```

# Network Updates Service

## Iteration 1

### ➤ Lessons learned:

- Centralizing updates into a single service leaves a single point of failure
- Be prepared to spend time tuning the `HttpConnectionManager` (timeouts, max connections)
- While the system was stabilizing, it was affecting all users; should have rolled the new service out to a small subset!
- Don't use "Least Frequently Used" (LFU) in a large `EHCache`—very bad performance!

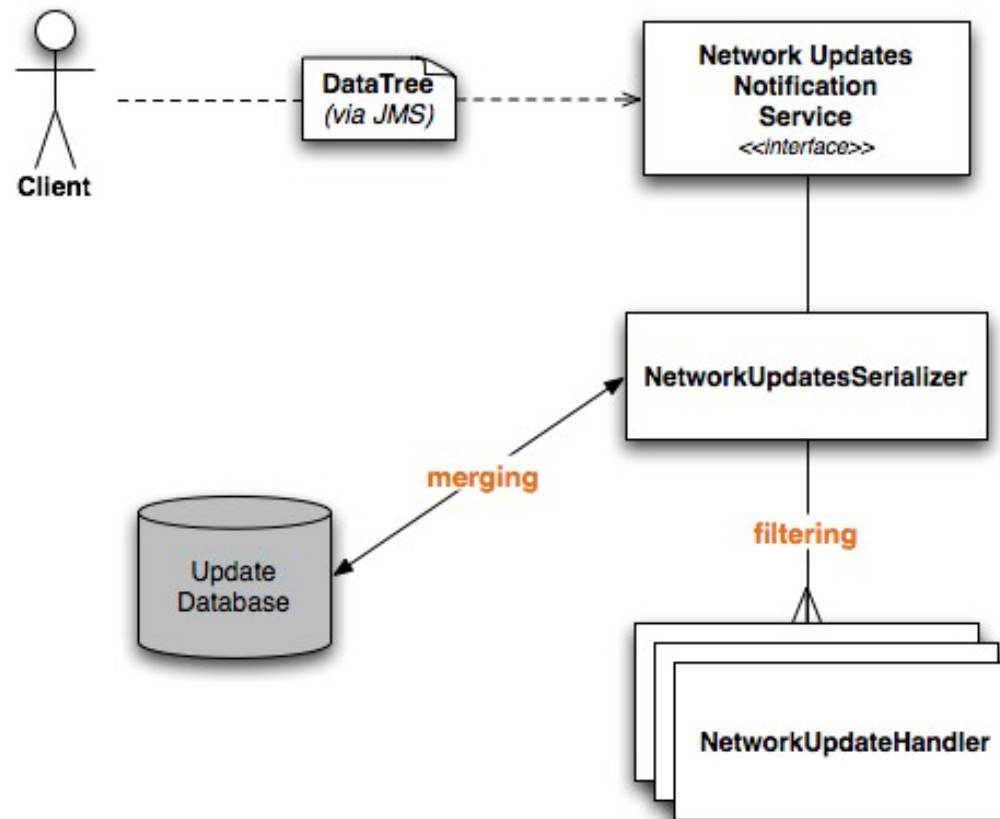
# Network Updates Service

## Iteration 2

- Hollywood Principle: *“Don’t call me, I’ll call you”*
- Push update when an event occurs
- Reading is much quicker since we don’t have to search for the data!
- Tradeoffs
  - Distributed updates may never be read
  - More storage space needed

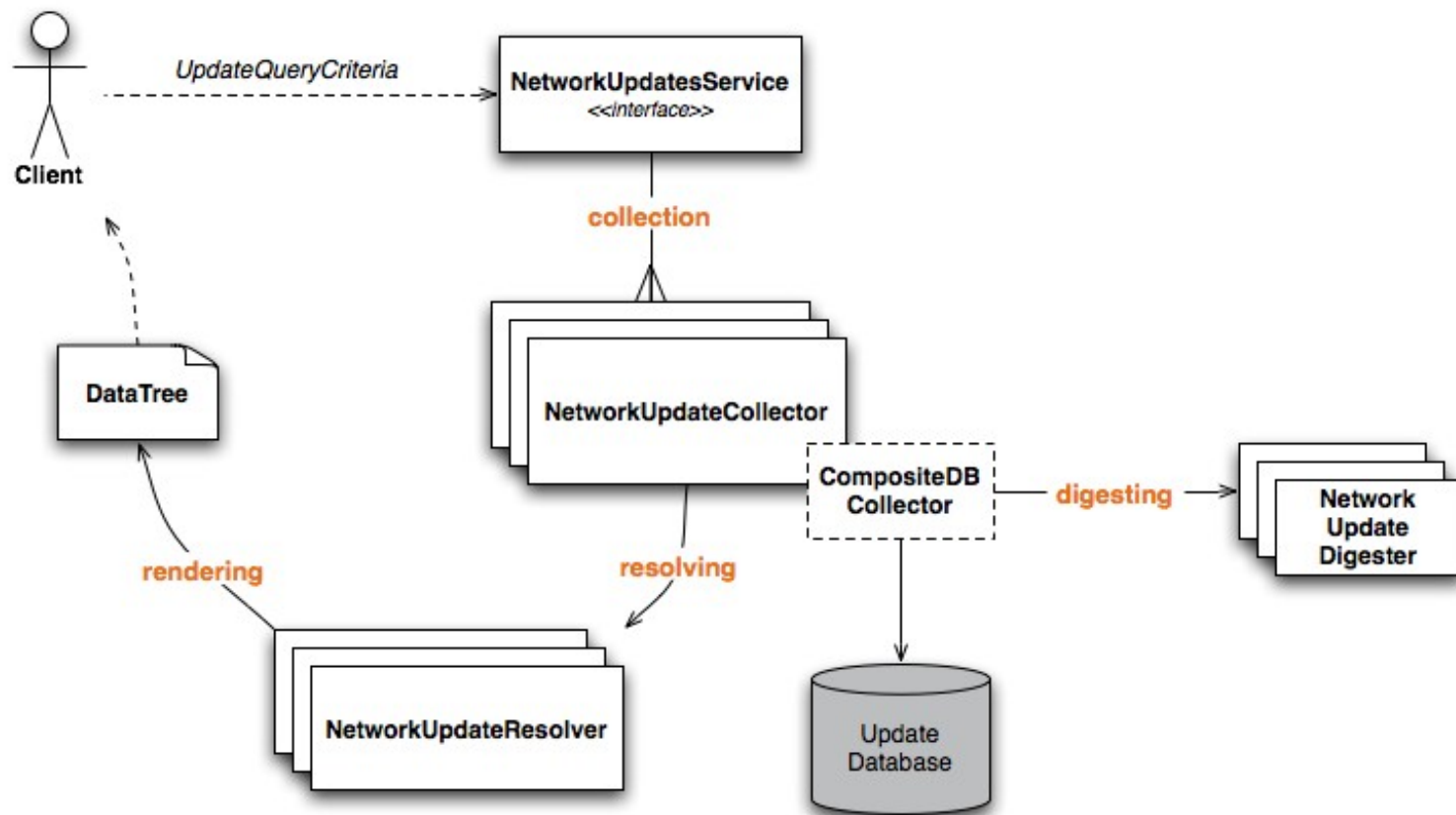
# Network Updates Service

## Iteration 2 - Pushing Events



# Network Updates Service

## Iteration 2 - Reading Updates



# Network Updates Service

## Iteration 2

### > Pushing Updates

- Updates are delivered via JMS
- Aggregate data stored in 1 CLOB column for each target user
- Incoming updates are merged into the aggregate structure using optimistic locking to avoid lock contention

### > Reading Updates

- Add a new collector that reads from the Update Database
- Use Digesters to perform arbitrary transformations on the stream of updates (e.g, collapse 10 updates from a user into 1)

# Network Updates Service

## Iteration 2

### ➤ Lessons learned:

- Underestimated the volume of updates to be processed
- CLOB block size was set to 8k, leading to a lot of wasted space (which isn't reclaimed!)
- Real-time monitoring/configuration with Java Management Extension (JMX™) specification was extremely helpful

# Network Updates Service

## Iteration 3

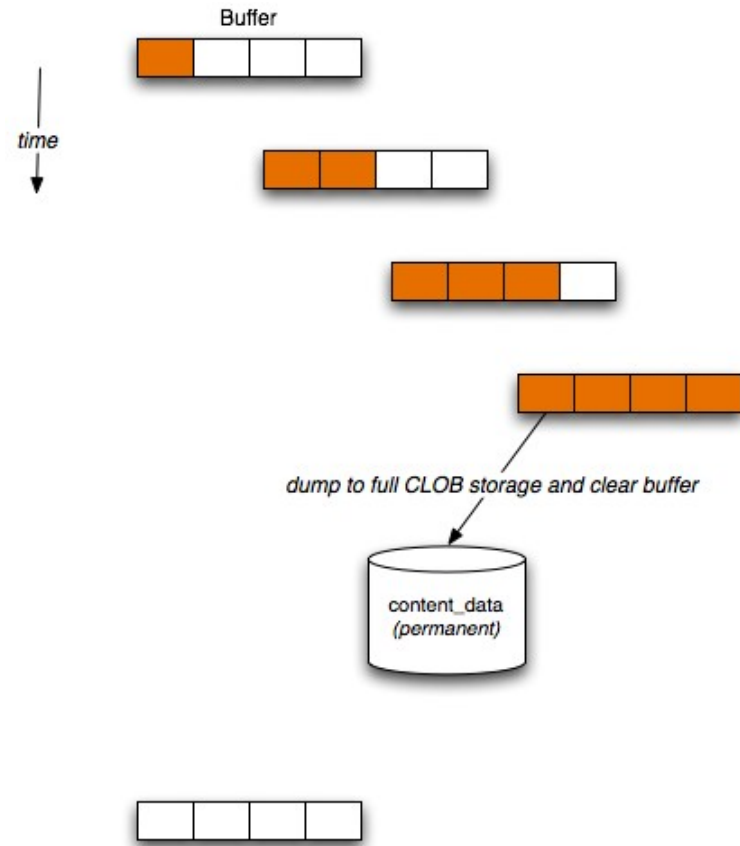
- Updating a CLOB is expensive
- Goal: Minimize the number of CLOB updates
  - Use an overflow buffer
  - Reduce the size of the updates



# Network Updates Service

## Iteration 3 - Overflow Buffer

- Add VARCHAR(4000) column that acts as a *buffer*
- When the buffer is full, dump it to the CLOB and reset
- Avoids over 90% of CLOB updates (depending on type), while still retaining the flexibility for more storage



# Scaling the system

## > What you learn as you scale:

- A single database does not work
- Referential integrity will not be possible
- Cost becomes a factor: databases, hardware, licenses, storage, power
- Any data loss is a problem
- Data warehousing and analytics becomes a problem
- Your system becomes a target for spamming exploits, data scraping, etc.

## > What to do:

- Partition everything:
  - by user groups
  - by domain
  - by function
- Caching is good even when it's only modestly effective
- Give up on 100% data integrity
- Build for asynchronous flows
- Build with reporting in mind
- Expect your system to fail at any point
- Never underestimate growth trajectory

# LinkedIn Communication Architecture

- Build with scalability in mind - never know when your business will take off
- Expect to do a lot of architecture and code refactoring as you learn and appreciate growth challenges

# THANK YOU



## LinkedIn Communication Architecture

Ruslan Belkin (<http://www.linkedin.com/in/rbelkin>)

Sean Dawson (<http://www.linkedin.com/in/seandawson>)

We are hiring!



# The Communication Service

## LinkedIn Spring Extensions

- Automatic context instantiation from multiple spring files
- LinkedIn Spring Components
- Property expansion
- Automatic termination handling
- Support for Builder Pattern
- Custom property editors:
  - Timespan (30s, 4h34m, etc.)
  - Memory Size, etc.

# The Communication Service

## LinkedIn Spring Extensions

```
Comm-server/  
  cmpt/  
    components/  
      ccsServiceExporter.spring  
      comm.spring  
      jmx.spring  
      comm-server.properties  
      corpMboxServiceExporter.spring  
      main.spring  
      comm-server.spring  
      memberMboxServiceExporter.spring  
      comm.properties  
      guestMboxServiceExporter.spring  
  build.xml  
impl/  
...
```

# The Communication Service

## LinkedIn Spring Extensions

```
...
<bean id="resolver"
class="com.linkedin.comm.pub.impl.MessageAddressResolver">
  <lin:config>
    <property name="resolverDB" ref="resolverDB"/>
    <property name="eos" ref="eos"/>
    <property name="els" ref="eos"/>
    <property name="memberAccessor"
ref="coreMemberAccessor"/>
  </lin:config>
</bean>
...
```

# The Communication Service

## LinkedIn Spring Extensions (Builder)

```
private final MessageAddressResolverDB _resolverDB;
...
MessageAddressResolver(Config config) { ... }
...
public static class Config    {
    private MessageAddressResolverDB _resolverDB;

    public MessageAddressResolverDB getResolverDB() {
        return ConfigHelper.getRequired(_resolverDB);
    }
    public void setResolverDB(MessageAddressResolverDB
                               resolverDB) {
        _resolverDB = resolverDB;
    }
} /*Config*/
```



# The Communication Service

## LinkedIn Spring Extensions (Components)

```
...  
<lin:component  
    id="remoteContentCommunicationService"  
    location="comm-server-client-cmpt">  
    <lin:wire property-name="activemq.producer.brokerURL"  
                property-  
value="${activemq.producer.brokerURL}"/>  
    <lin:wire property-name="comm.server.httpRpc.url"  
                property-  
value="${leo.comm.server.httpRpc.url}"/>  
</lin:component>  
...
```