



jboss
.org

Drools Complex Event Processing with Twitter4J

Toshiya Kobayashi

Red Hat Senior Software Maintenance Engineer

Agenda

- Overview
- Simple Demo
- Guvnor Integration
- Developing a bot

Agenda

- Overview
- Simple Demo
- Guvnor Integration

Wifi

SSID : Judcon2012-6

pass : 6judcon2012

Overview



Knowlege Base
with STREAM mode

Rule1

Rule2

Rule3

Match?
Fire!

Knowlege Session



Twitter Stream API



Tweet

Tweet

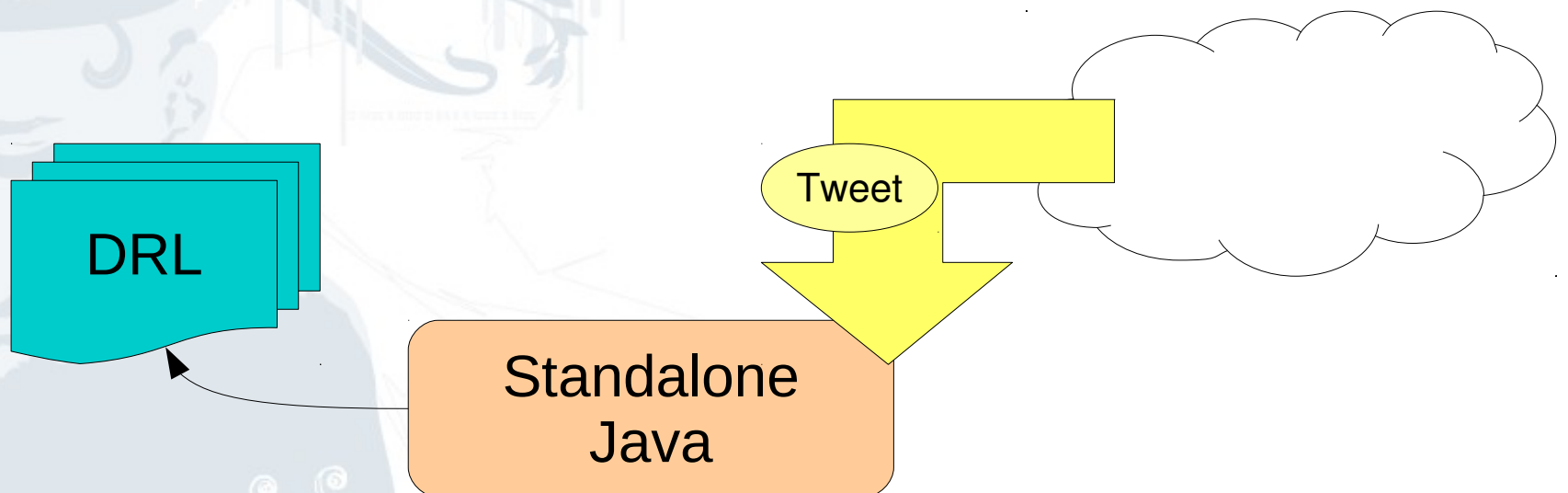
Tweet

EntryPoint "twitter"

Tweet

Simple Demo

- Command Line Interface with Eclipse
- Static rules
- Good for understanding the basic concept



Simple Demo

- <https://github.com/droolsjbpm/droolsjbpm-contributed-experiments>
 - original
 - contains this 'Simple Demo' only
- <https://github.com/tkobayas/droolsjbpm-contributed-experiments>
 - fork
 - judcon branch contains 'Guvnor integration' and 'Bot' as well

Simple Demo

```
$ git clone git://github.com/tkobayas/droolsjbpm-  
contributed-experiments.git  
$ cd droolsjbpm-contributed-experiments  
$ git checkout --track origin/judcon  
$ cd twittercbr/  
$ mvn clean package
```

Drools Complex Event Processing

- <http://www.jboss.org/drools/drools-fusion.html>
- Known as Drools Fusion
- "Complex Event Processing, or CEP, is primarily an event processing concept that deals with the task of processing multiple events with the goal of identifying the meaningful events within the event cloud. CEP employs techniques such as detection of complex patterns of many events, event correlation and abstraction, event hierarchies, and relationships between events such as causality, membership, and timing, and event-driven processes."

Drools CEP

- Declare Event in Rule

```
declare Status
  @role( event )
  @timestamp( createdAt )
  @expires( 2s )
end

rule "Dump tweets"
when
  $s : Status() from entry-point "twitter"
then
  System.out.println( ... );
end
```

Drools CEP

- Declare Event in Rule

```
declare Status
    @role( event )
    @timestamp( createdAt )
    @expires( 2s )
end

rule "Dump tweets"
when
    $s : Status() from entry-point "twitter"
then
    System.out.println( ... );
end
```

Drools CEP

- Declare Event in Rule

```
declare Status
  @role( event )
  @timestamp( createdAt )
  @expires( 2s )
end

rule "Dump tweets"
when
  $s : Status() from entry-point "twitter"
then
  System.out.println( ... );
end
```

Drools CEP

- Declare Event in Rule

```
declare Status
  @role( event )
  @timestamp( createdAt )
  @expires( 2s )
end

rule "Dump tweets"
when
  $s : Status() from entry-point "twitter"
then
  System.out.println( ... );
end
```

Drools CEP

- Declare Event in Rule

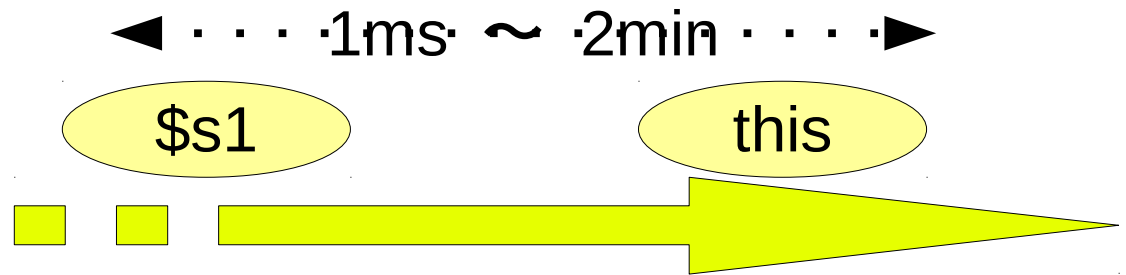
```
declare Status
  @role( event )
  @timestamp( createdAt )
  @expires( 2s )
end

rule "Dump tweets"
when
  $s : Status() from entry-point "twitter"
then
  System.out.println( ... );
end
```


Drools CEP

- Temporal operators

```
rule "Dump tweets from user conversation"  
when  
    $s1 : Status( $name : user.screenName ) from entry-point "twitter"  
    $s2 : Status( inReplyToScreenName == $name,  
        this after[1ms,2m] $s1 ) from entry-point "twitter"  
then  
    System.out.println( ... );  
end
```



Drools CEP

- Set STREAM mode

```
KnowledgeBaseConfiguration conf =  
KnowledgeBaseFactory.newKnowledgeBaseConfiguration();  
  
conf.setOption( EventProcessingOption.STREAM );  
  
KnowledgeBase kbase =  
KnowledgeBaseFactory.newKnowledgeBase( conf );
```

Drools CEP

- Create an Entry Point and insert events into it in a different thread

```
final WorkingMemoryEntryPoint ep =  
ksession.getWorkingMemoryEntryPoint( "twitter" );  
...  
  
public void onStatus( Status status ) {  
    ep.insert( status );  
}
```

Drools CEP

- Fire until halt

```
ksession.fireUntilHalt();
```

Twitter4J

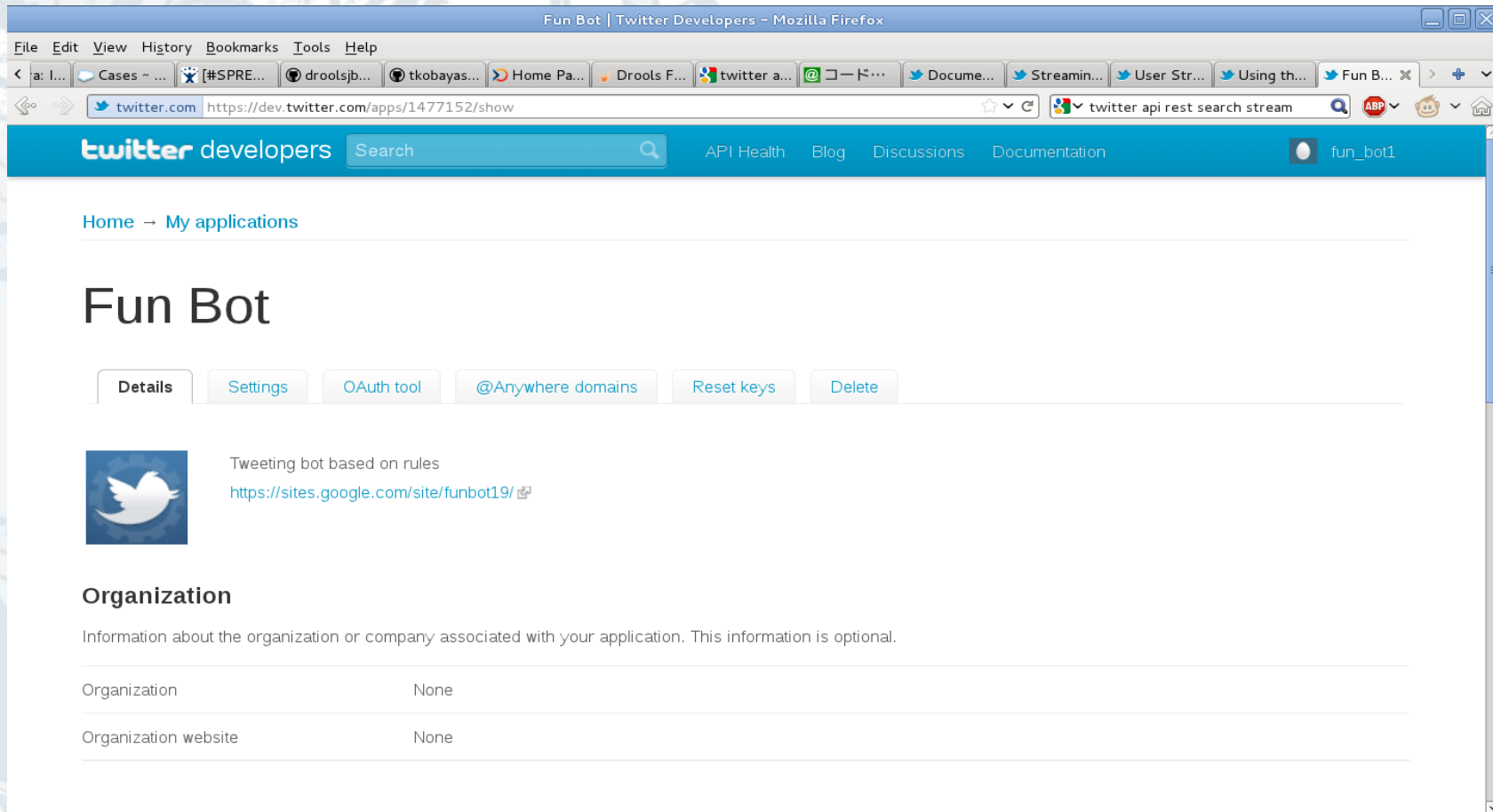
- <http://twitter4j.org>
- Java library for the Twitter API
 - REST
 - Search
 - Stream

Twitter4J

- Stream API
 - Streamed Tweets
 - sample
 - filter
 - firehose
 - links
 - retweet
 - User Stream
 - Site Stream

Twitter4J

- Register to <https://dev.twitter.com/> to acquire your access token



The screenshot shows a web browser window titled 'Fun Bot | Twitter Developers - Mozilla Firefox'. The address bar shows the URL 'https://dev.twitter.com/apps/1477152/show'. The page header includes 'twitter developers' and a search bar. The main content area shows the application name 'Fun Bot' and a navigation bar with buttons for 'Details', 'Settings', 'OAuth tool', '@Anywhere domains', 'Reset keys', and 'Delete'. Below this, there is a Twitter logo and the text 'Tweeting bot based on rules' with a link to 'https://sites.google.com/site/funbot19/'. The 'Organization' section is visible, with fields for 'Organization' and 'Organization website', both currently set to 'None'.

Twitter4J

- Edit twitter4j.properties to enable OAuth

```
oauth.consumerKey=XXXXXXXXXXXXXXXXXX  
oauth.consumerSecret=XXXXXXXXXXXXXXXXXX  
oauth.accessToken=XXXXXXXXXXXXXXXXXX  
oauth.accessTokenSecret=XXXXXXXXXXXXXXXXXX
```

Twitter4J

- Get a `TwitterStream` instance, add a custom listener and read a sample stream

```
StatusListener listener = new TwitterStatusListener( ep );  
TwitterStream twitterStream = new  
TwitterStreamFactory().getInstance();  
TwitterStream.addListener( listener );  
twitterStream.sample();
```

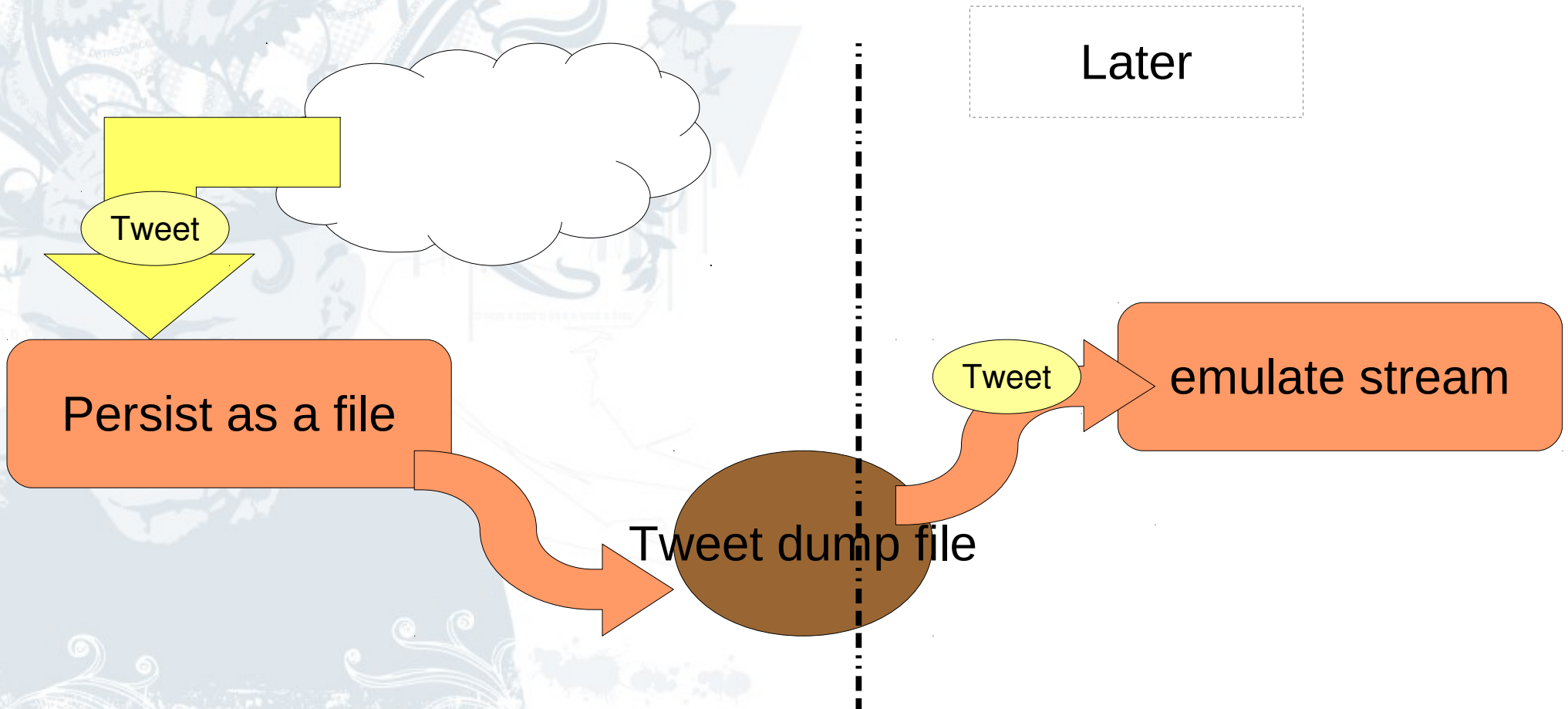
Twitter4J

- A custom listener

```
public class TwitterStatusListener implements StatusListener {  
...  
    public void onStatus( Status status ) {  
        ep.insert( status );  
    }  
...  
}
```


Twitter4J

- Offline stream is useful in case that Twitter/network is unavailable



Twitter4J

- Offline stream is useful in case that Twitter/network is unavailable

```
private static void feedEvents( final StatefulKnowledgeSession ksession,
                               final WorkingMemoryEntryPoint ep ) {
    try {
        StatusListener listener = new TwitterStatusListener( ep );
        ObjectInputStream in = new ObjectInputStream( new
        FileInputStream( "src/main/resources/twitterstream.dump" ) );
        SessionPseudoClock clock = ksession.getSessionClock();

        for( int i = 0; ; i++ ) {
            try {
                Status st = (Status) in.readObject();
                clock.advanceTime( st.getCreatedAt().getTime() - clock.getCurrentTime(),
                TimeUnit.MILLISECONDS );
                listener.onStatus( st );
            } catch( IOException ioe ) {
                break;
            }
        }
    }
}
```

Simple Demo

- DEMO

Wifi

SSID : Judcon2012-6

pass : 6judcon2012

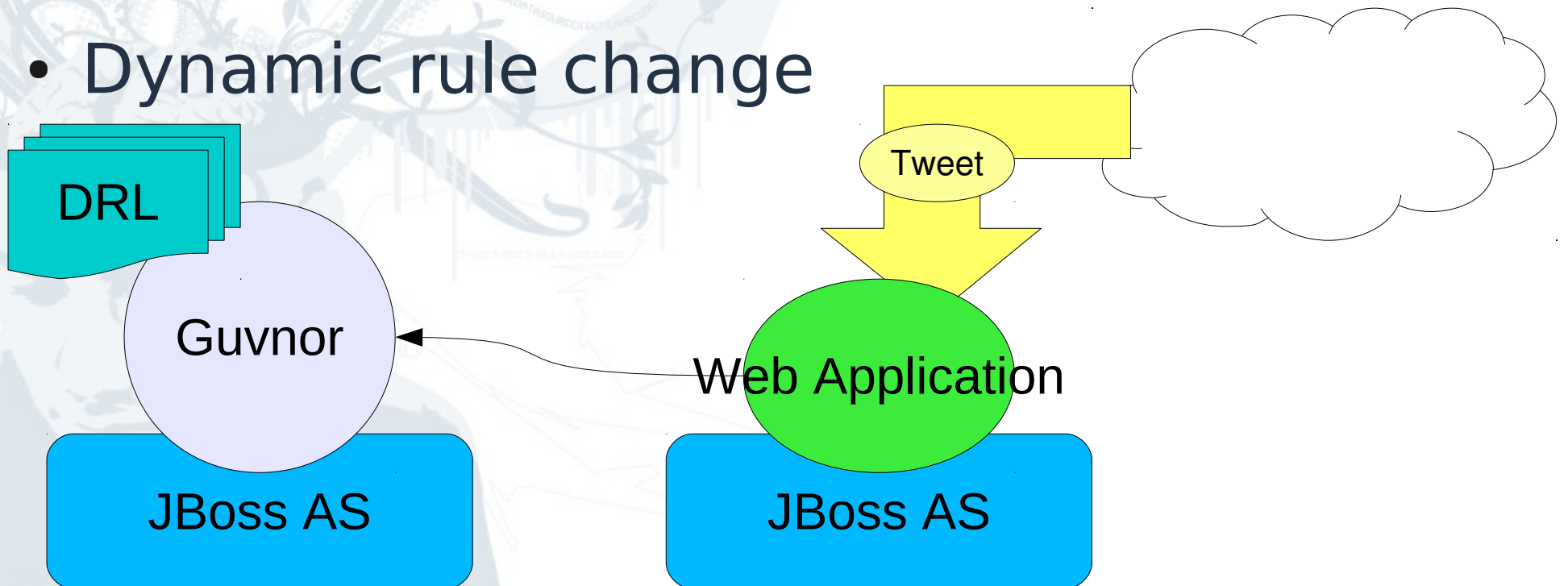
Twitter4J

- Filter stream

```
TwitterStream twitterStream = new  
TwitterStreamFactory().getInstance();  
twitterStream.addListener( listener );  
FilterQuery query = new FilterQuery();  
query.track(new String[]{"#judcon"});  
twitterStream.filter(query);
```

Guvnor Integration

- Authoring rules with Guvnor GUI Editor
- Simple Web Application which re-loads results with Ajax poll
- Dynamic rule change



Guvnor

- <http://www.jboss.org/drools/drools-guvnor.html>
- A centralized repository for Drools Knowledge Bases, with rich web based GUIs




The screenshot displays the JBoss Guvnor web interface. The browser window title is "JBoss Guvnor - Mozilla Firefox". The address bar shows the URL: "localhost:8080/guvnor-5.3.2-SNAPSHOT-jboss-as-5.1/org.drools.guvnor.Guvnor/Guvnor.html#ModuleEditorPlace:95efd46c...". The page title is "Drools". The main content area shows a configuration editor for a package named "org.drools.examples.twittercbr". The configuration includes "Imported types" (twitter4j.Tweet, twitter4j.Status, twitter4j.Place, twitter4j.User) and "Globals" (List [tweetList]). There are radio buttons for "Build whole package", "Use built-in selector", and "Use custom selector". Below the configuration, there are links for package documentation, source, binary, and running tests.




Guvnor

- Develop rules with CEP support

WHEN

All Status [\$s1] with:

1. user   

 [\$name] User.screenName   

Over sliding window

From Entry Point

All Status [\$s2] with:

2. inReplyToScreenName equal to

 this after

Over sliding window

From Entry Point

THEN

`System.out.println(`

Note: There are some issues in Editor with CEP. Please check the latest build.

Guvnor

- <https://github.com/droolsjbpm/guvnor/>
 - original
- <https://github.com/tkobayas/guvnor/>
 - fork
 - judcon branch contains quick fixes for some rule editor CEP issues. I will raise JIRAs soon ;)

```
$ git clone git://github.com/tkobayas/guvnor.git  
$ cd guvnor  
$ git checkout --track origin/judcon  
$ mvn clean package -DskipTests
```

Note: There are some issues in Editor with CEP. Please check the latest build.

Knowledge Agent

- Knowledge Agent supports dynamic rule loading from Guvnor repository

```
KnowledgeBaseConfiguration kbaseConf = KnowledgeBaseFactory
    .newKnowledgeBaseConfiguration();
kbaseConf.setOption(EventProcessingOption.STREAM);
KnowledgeBase kbase =
KnowledgeBaseFactory.newKnowledgeBase(kbaseConf);
kagent = KnowledgeAgentFactory.newKnowledgeAgent("MyAgent",
kbase);
kagent.applyChangeSet(ResourceFactory
    .newClassPathResource("ChangeSet.xml"));
```

Note: The KnowledgeBaseConfiguration will be re-used when the knowledge agent re-build a knowledge base. So Stream mode will be applied continuously

Knowledge Agent

- ResourceChangeScanner scans the repository with the configured interval

```
ResourceChangeScannerConfiguration sconf = ResourceFactory
    .getResourceChangeScannerService()
    .newResourceChangeScannerConfiguration();
sconf.setProperty("drools.resource.scanner.interval", "20");
ResourceFactory.getResourceChangeScannerService().configure(s
conf);
```

```
ResourceFactory.getResourceChangeScannerService().start();
ResourceFactory.getResourceChangeNotifierService().start();
```

Note: Use the same Drools version in Guvnor and the client

Knowledge Agent

- Use KnowledgeAgentEventListener to trigger action on resource change

```
....
    kagent.addEventListener(new HaltKnowledgeAgentEventListener());
....

static class HaltKnowledgeAgentEventListener extends
    DefaultKnowledgeAgentEventListener {
    public void afterChangeSetApplied(AfterChangeSetAppliedEvent event) {
        // If the changeset is updated, halt the ksession and re-run
        if (ksessionRunning) {
            try {
                stopKnowledgeSession();
                run();
            } catch (Exception e) {
                ....
            }
        }
    }
}
```

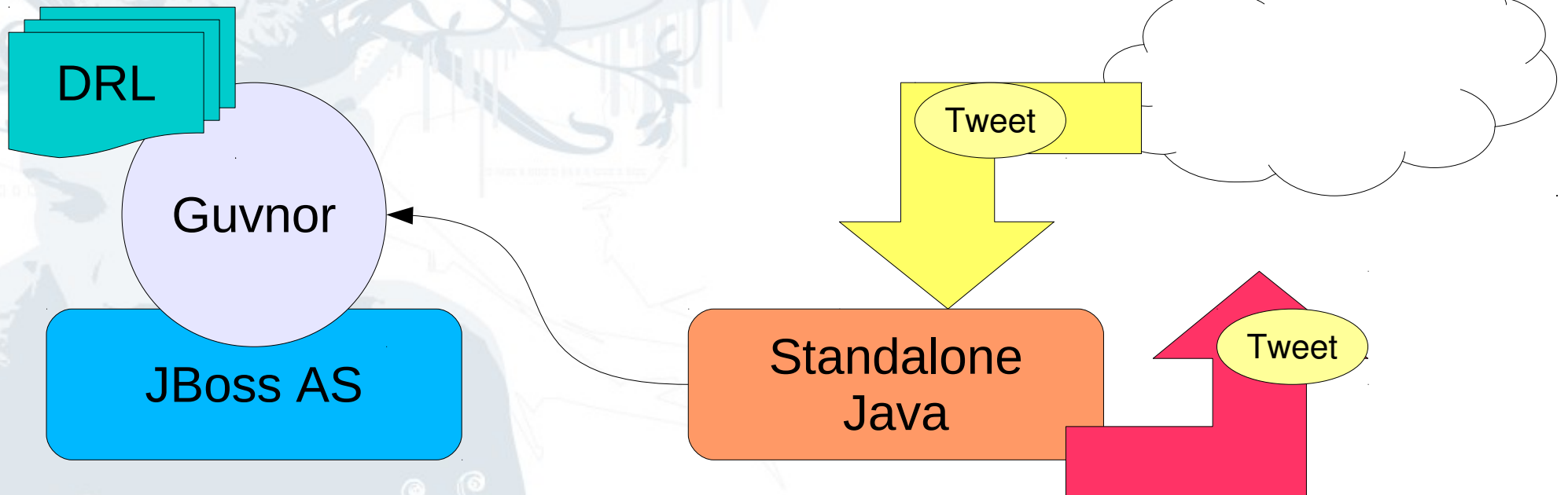
Guvnor Integration



- DEMO

Developing a bot

- Have fun
- Do not spam



Developing a bot



Scolding Bot (@scolding_bot)

107 TWEETS

0 FOLLOWING

8 FOLLOWERS

https://twitter.com/#!/scolding_bot

great.a
← In reply to Toshiya Kobayashi

Developing a bot

- Which stream (filter/user/sample) do you use?
- How much do you narrow down the stream by FilterQuery?
- How to deal with rule update
 - What condition can be hard-coded?
- How to maintain the daemon thread

Please follow @scolding_bot
https://twitter.com/#!/scolding_bot

Developing a bot

- Too frequent login will be restricted
- Too frequent tweet will be restricted
- Duplicate tweet will be restricted

Please follow @scolding_bot
https://twitter.com/#!/scolding_bot

Developing a bot

- DEMO

Please follow @scolding_bot
https://twitter.com/#!/scolding_bot