MarkLogic

Teaching an Old Dog New Tricks: Agile Data Modeling & Developer Journey with MarkLogic NoSQL Database

Niko Schmuck, Jochen Jörg

© COPYRIGHT 2015 MARKLOGIC CORPORATION. ALL RIGHTS RESERVED.

MarkLogic

Motivation

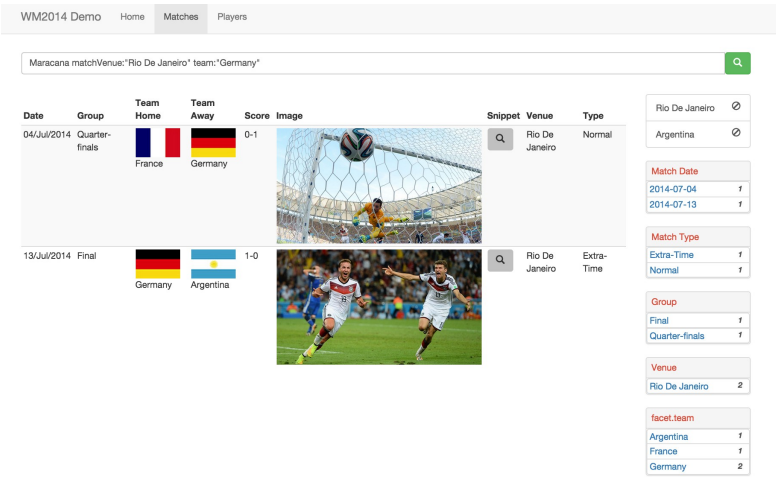
- Proof Thesis:
  "NoSQL increases flexibility and agility in the data layer"
  - Show case for Agile Data Management
  - Analytics Application on top of complex data with MarkLogic as NoSQL Technology
  - Developer Journey & Architecture

© COPYRIGHT 2015 MARKLOGIC CORPORATION. ALL RIGHTS RESERVED.

Application “WM14”

- Football World Cup Brazil 2014
- Finding out insights about
  - Matches
  - Teams
  - Players

  … based on heterogeneous sports data



© COPYRIGHT 2015 MARKLOGIC CORPORATION. ALL RIGHTS RESERVED.

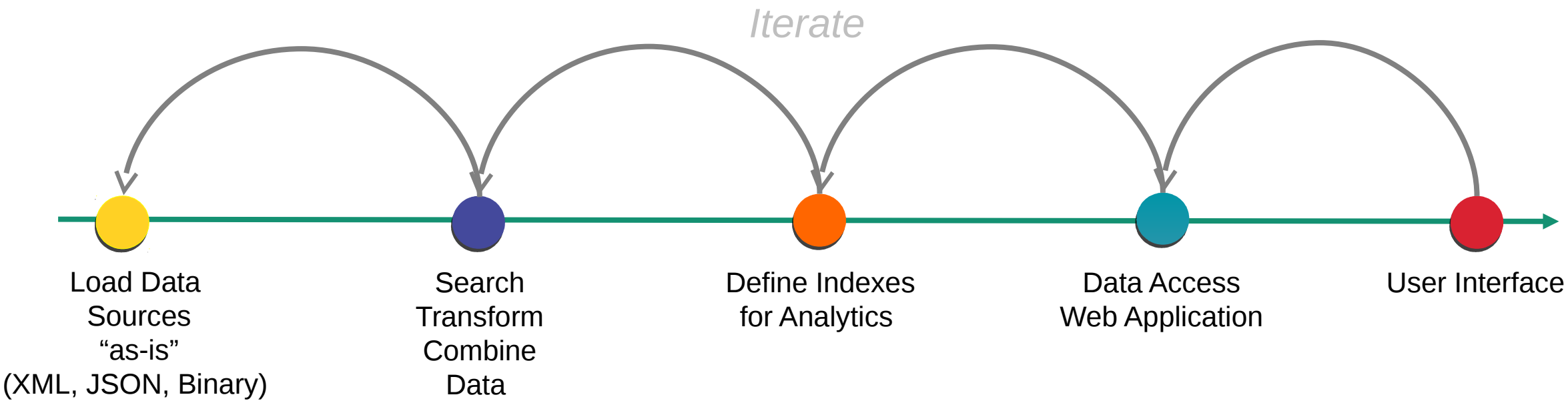
Frameworks & Tools

- MarkLogic NoSQL Platform Search & Store
- Spring MVC, Spring Boot
- Thymeleaf (Templating)
- Gradle: as build tool, but also
  for managing MarkLogic setup/deployment
- Apache Camel:
  - Integration Framework
  - Swiss Army Knife of consuming variety of data sources

© COPYRIGHT 2015 MARKLOGIC CORPORATION. ALL RIGHTS RESERVED.

Demo I: Introduce core entities

Developer Journey == Agile Process



© COPYRIGHT 2015 MARKLOGIC CORPORATION. ALL RIGHTS RESERVED.

Domain Model in the Middle Tier?

Generic  **Typed**

- Exposes raw JSON to UI as stored in database
- Shifts responsibility of Validation & Resilience rgd. sparse / data variety to UI

- Transform from/to JSON (semi / automatic mapping)
- Validation on ingestion
- Further access on domain objects (i.e. on templates)

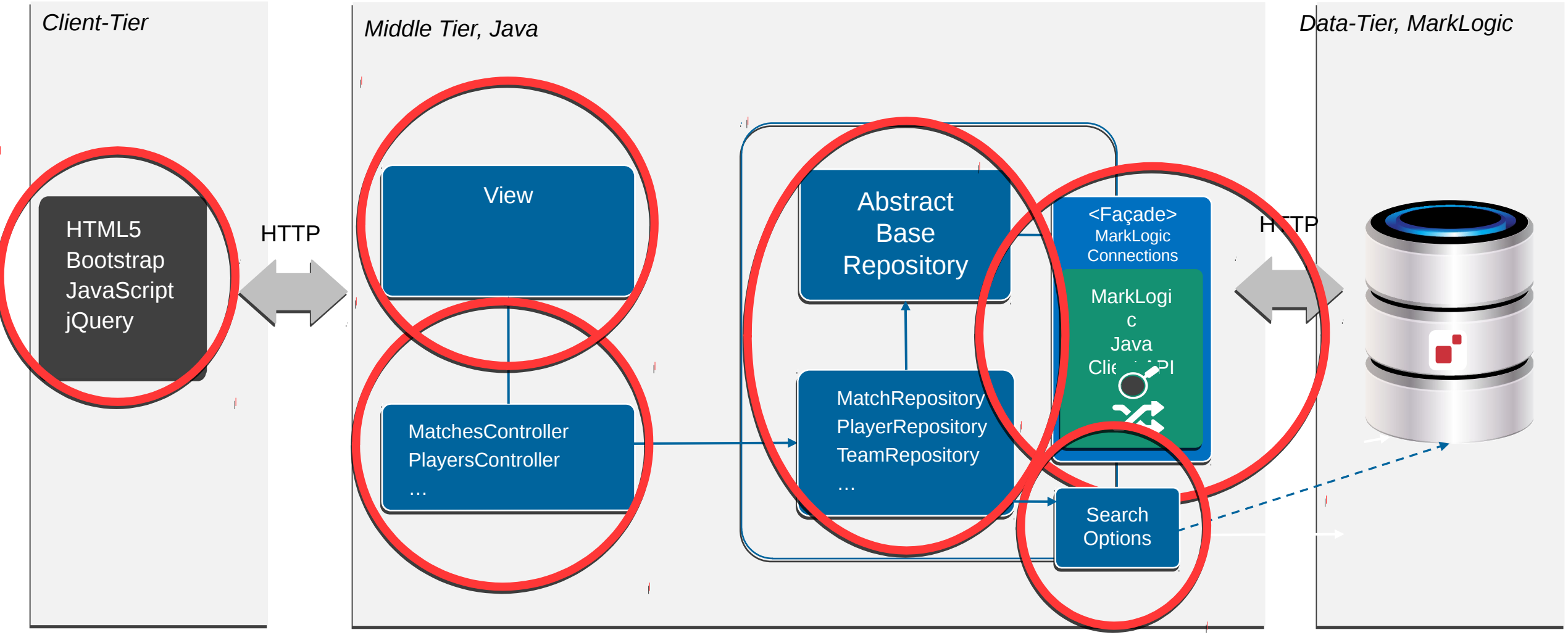
Hybrid

- Cover dynamic attributes
- (extend from XML Document)

© COPYRIGHT 2015 MARKLOGIC CORPORATION. ALL RIGHTS RESERVED.

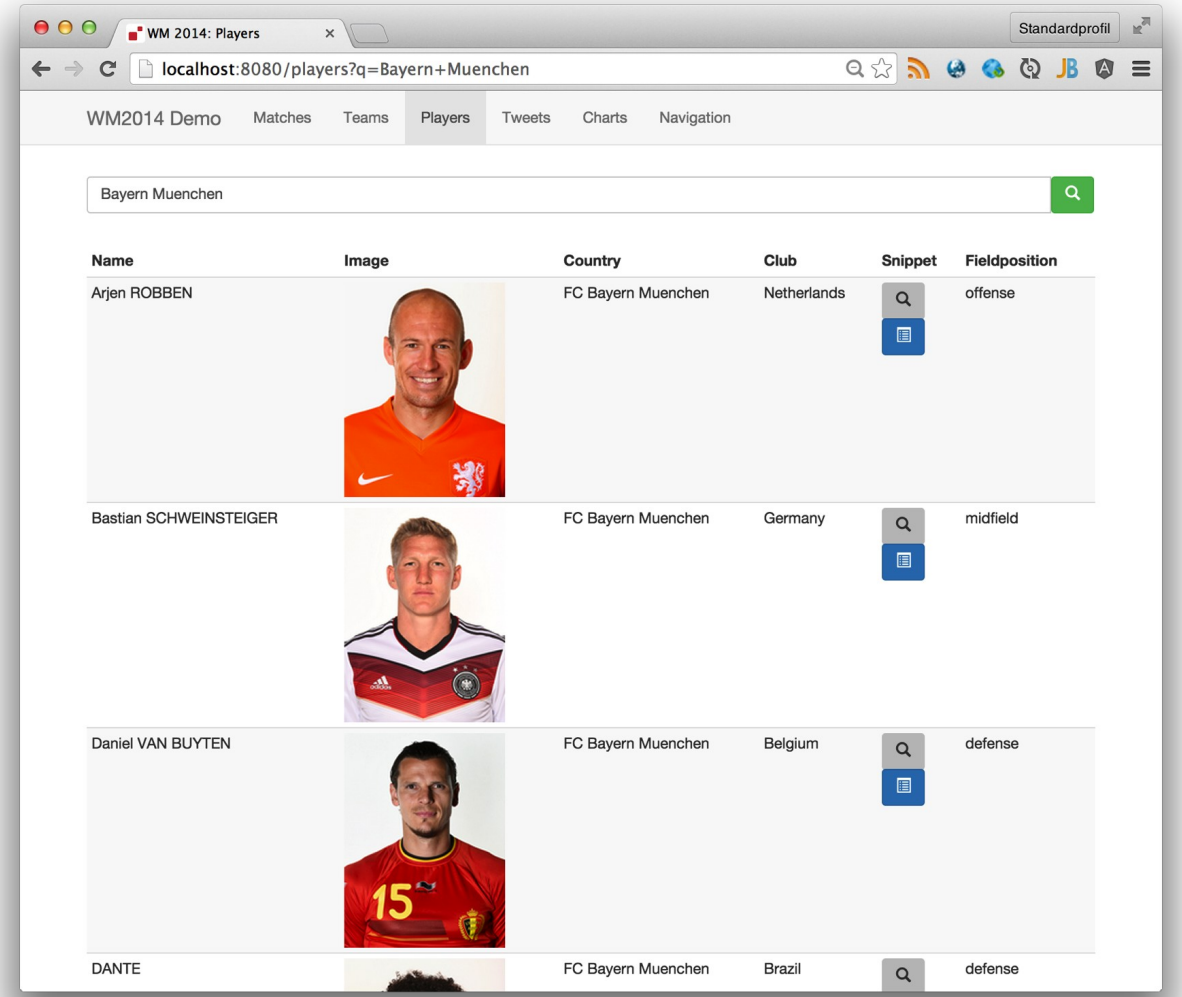
Demo II: Search scenarios

Application Architecture – Main components



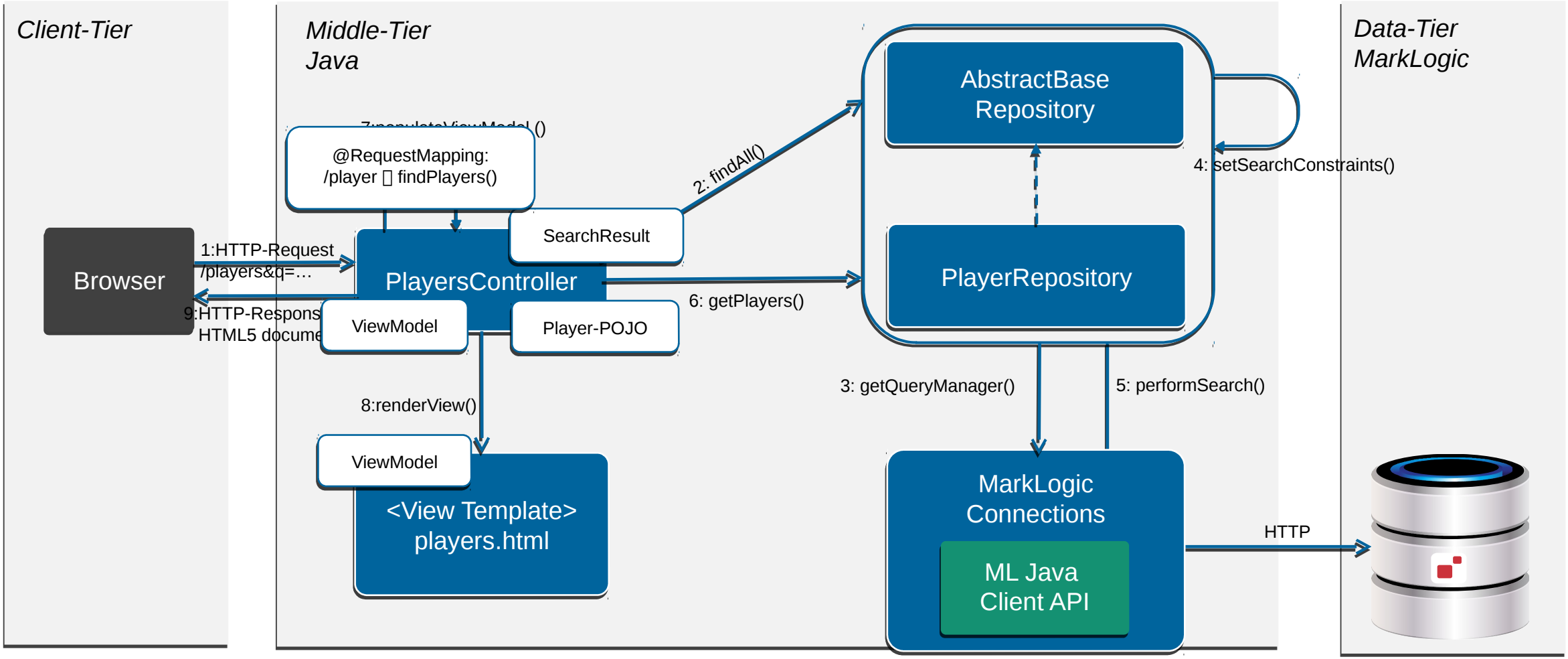
© COPYRIGHT 2015 MARKLOGIC CORPORATION. ALL RIGHTS RESERVED.

Request Lifecyle: Search for Players



© COPYRIGHT 2015 MARKLOGIC CORPORATION. ALL RIGHTS RESERVED.

Request Lifecyle: Search for Players



© COPYRIGHT 2015 MARKLOGIC CORPORATION. ALL RIGHTS RESERVED.

Controller

```
@Controller
public class PlayersController extends AbstractBaseController {

    @Autowired
    private PlayerRepository playerRepository;

    @RequestMapping(value = "/players”, method = RequestMethod.GET)
    public String findPlayers(@RequestParam(required = false, defaultValue = "") String q,
                              @RequestParam(required = false, defaultValue = "1") int page,
                              Map<String, Object> model) {
       SearchHandle searchResult = playerRepository.findAll(defaultUser, q, page);
       …
       model.put("facets", searchResult.getFacetResults());
       // map generic search result to domain objects
       model.put("resultItems", playerRepository.getResultObjects(defaultUser, searchResult));

       return "search/players";
    }

    …
}
```

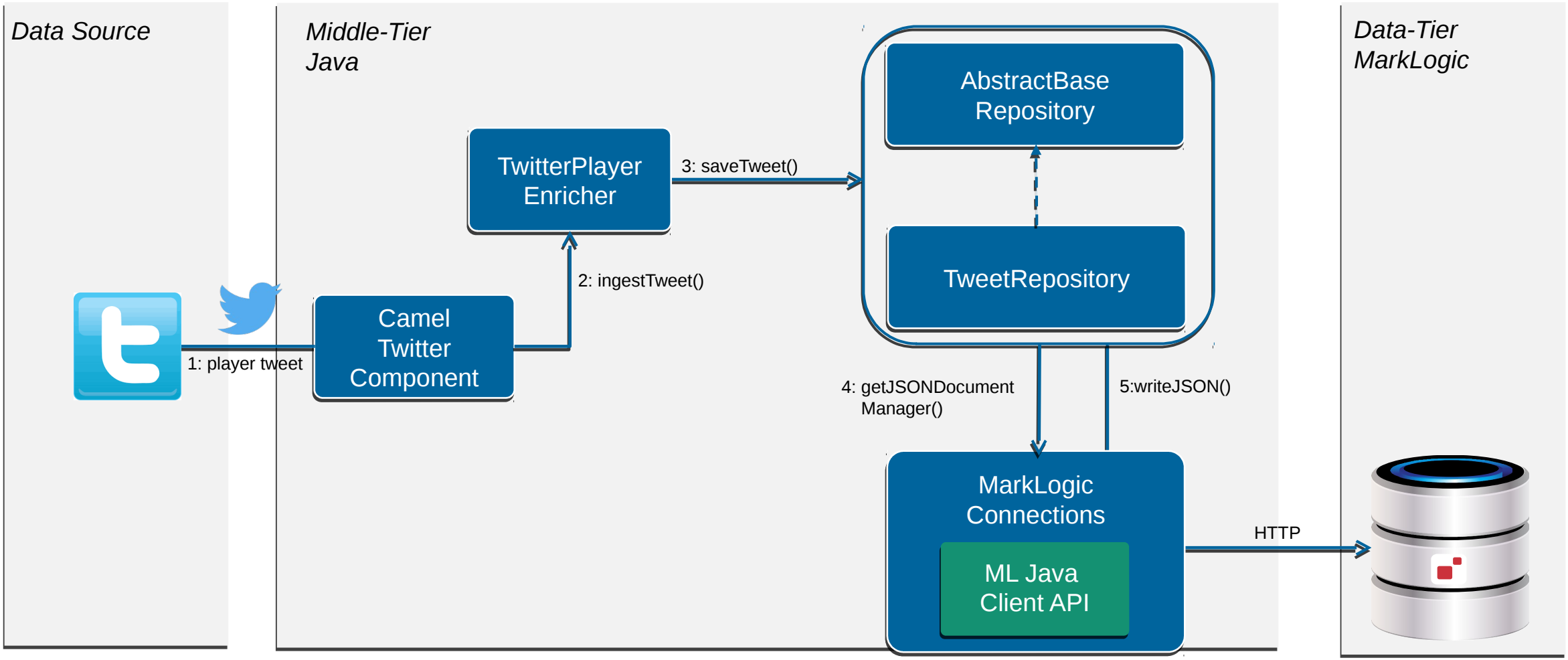
```
@Service
public abstract class AbstractBaseRepository<T> {
   @Autowired
   protected MarkLogicConnections connections;


   public SearchHandle findAll(String username, String query, int page) {
      long start = (page - 1) * defaultPageLength + 1;
      QueryManager queryManager = connections.getQueryManager(username);
      StringQueryDefinition qdef = queryManager.newStringDefinition(getOptionsName());
      qdef.setCriteria(query);
      qdef.setDirectory(getDocumentDirectory());
      SearchHandle resultsHandle = new SearchHandle();
      queryManager.search(qdef, resultsHandle, start);
      return resultsHandle;
   }

}
```

MarkLogic

Real-time Integration of Twitter Streaming API



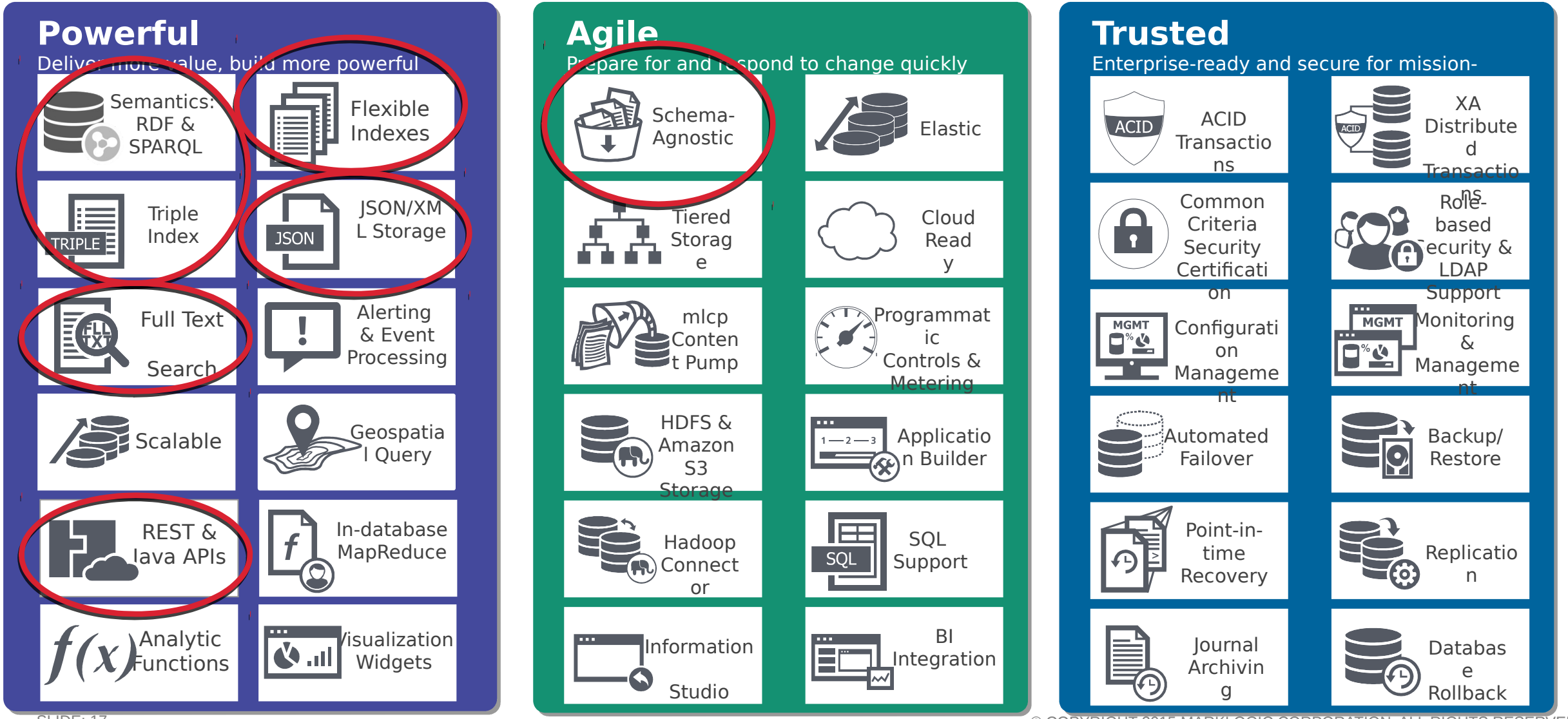
© COPYRIGHT 2015 MARKLOGIC CORPORATION. ALL RIGHTS RESERVED.

```
@Bean
public RouteBuilder routeBuilder() {
    return new TwitterRouteBuilder();
}
private class TwitterRouteBuilder extends RouteBuilder {
    @Override
    public void configure() throws Exception {
        // setup Twitter component
        TwitterComponent tc = getContext().getComponent("twitter”,TwitterComponent.class);
        tc.setAccessToken(twitter.getConfiguration().getOAuthAccessToken());
        tc.setAccessTokenSecret(twitter.getConfiguration().getOAuthAccessTokenSecret());
        tc.setConsumerKey(twitter.getConfiguration().getOAuthConsumerKey());
        tc.setConsumerSecret(twitter.getConfiguration().getOAuthConsumerSecret());

        // Follow real-time tweet stream for all players ...
        String userIds = StringUtils.arrayToDelimitedString(
                  TwitterPlayerEnricher.TWITTER_USER_IDS.toArray(), ",");
        from("twitter://streaming/filter?type=event&userIds=" + userIds)
            .beanRef("twitterPlayerEnricher", "ingestTweet")
            .log("~~~~ Tweet ${body}");
    }
}
```

Wrap-up

WM14 – Role of MarkLogic



© COPYRIGHT 2015 MARKLOGIC CORPORATION. ALL RIGHTS RESERVED.

MarkLogic

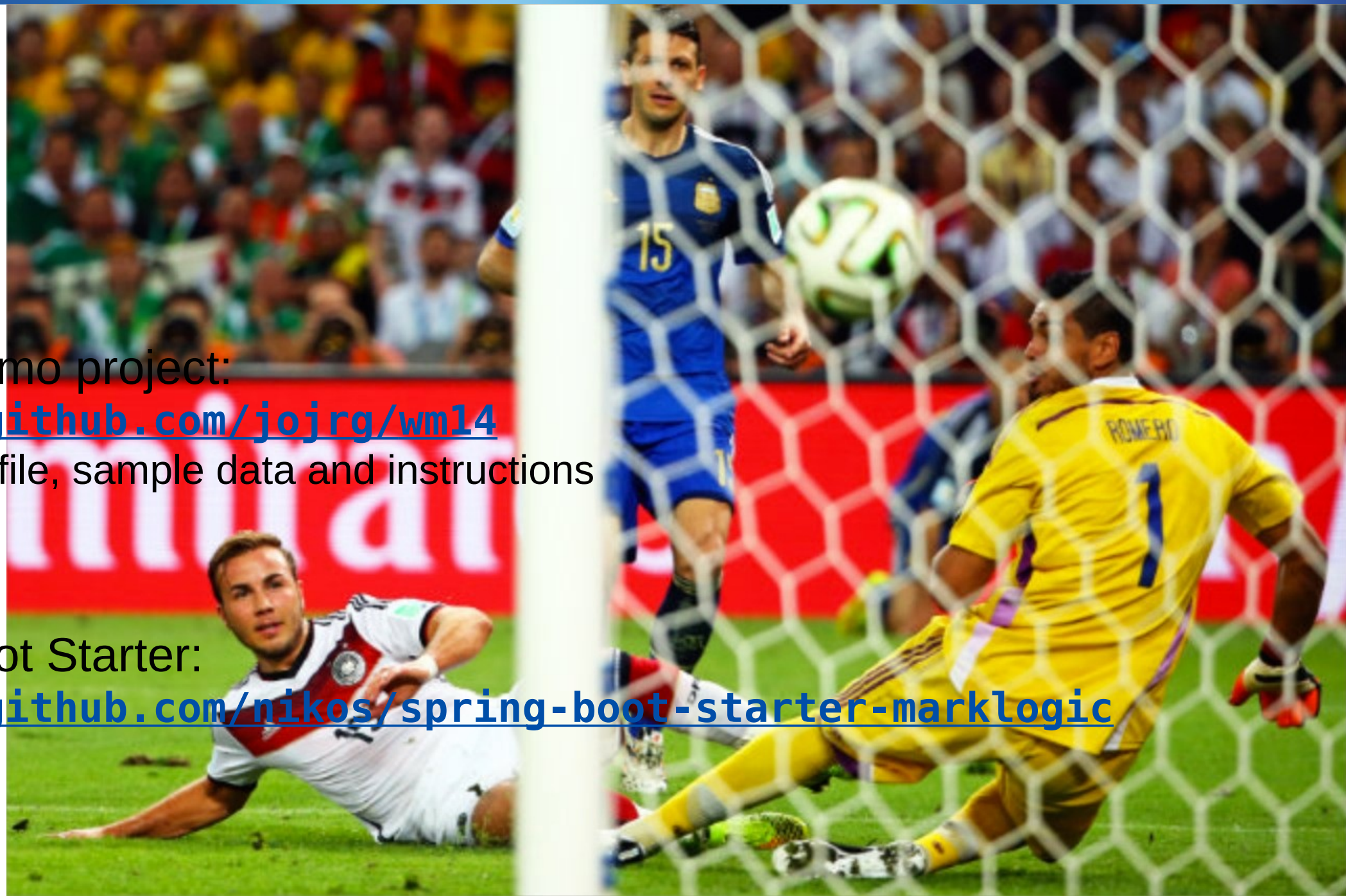
Conclusion and Take away

- Agile data modeling with NoSQL
  - Adapt to changes
  - Tackle moving targets
- Separation of concerns
  - Maintainable code base
  - Supports cross functional teams
  - Architecture evolves with your business

© COPYRIGHT 2015 MARKLOGIC CORPORATION. ALL RIGHTS RESERVED.

MarkLogic

Join the



WM14 Demo project:
https://github.com/jojrg/wm14
incl. Dockerfile, sample data and instructions

Spring Boot Starter:
https://github.com/nikos/spring-boot-starter-marklogic

MarkLogic

Danke!

Niko Schmuck

n.schmuck@nava.de

@niko_nava

de.linkedin.com/in/nikoschmuck

Jochen Jörg

jochen.joerg@marklogic.com

@jochenjoerg

de.linkedin.com/in/jochenjoerg

© COPYRIGHT 2015 MARKLOGIC CORPORATION. ALL RIGHTS RESERVED.