KAIZEN
Software

Player Portfolio Management System

Software Development Proposal

Technical Approach

Solicitation No:

7/1/2017

Cover Letter

7/1/2017

IdeaSports

Attention: Eduardo Friscaro

Subject: Proposal from Kaizen Software Technologies for Development of the Player Portfolio Management Software Application System

IdeaSports can gain optimal results by selecting an adaptive IT company like Kaizen Software Technologies with the ability of providing integrated solutions that significantly reduce the time and risk associated with designing and implementing IT solutions. We use small effective project teams that collaborate with our customers through the phases of requirements gathering, design, and realization of the working product. Our approach enhances Return on Investment by focusing on a user friendly and mission oriented end product. Our solution offers:

- Complete, secure, and thoroughly tested software deliverables
- Detailed documentation of all deliverables to enable independent third parties to install and manage the hosting of the software
- An appropriate mix of personnel to meet all requirements

Kaizen provides end-to-end DevOps services that deliver the expertise, skills, solutions and quality required to successfully scope, design, and deliver enterprise applications.

Kaizen will honor the services identified in its proposal. Inquiries may be addressed to my attention at 407-353-6642 or denver.robert.williams@gmail.com.

Sincerely,

Dr. Denver R.E. Williams
President, Kaizen Software Technologies

Table of Contents

1 Executive Summary

Kaizen Software Technologies (Kaizen) is a small minority-owned, EDWOSB technology business that is CMMI Level-3 appraised for efficient development and is also ISO 9001:2015 certified for our Quality Management Processes. Kaizen has developed our expertise over years of providing end to end O&M support for large mission critical systems for various clients including U.S. Patent and Trademark Office (USPTO). Our experience includes design, development, deployment, maintenance, and modernization of many of the business and mission critical systems of our clients.

Kaizen is a leader in Agile Methodology implementation and training, systems engineering, and business process reengineering with extensive experience in developing and maintaining both small and large mission critical systems leveraging Java, SharePoint, Oracle, MS SQL Server, Cloud, and distributed Java technology stack. Kaizen balances risk mitigation and innovative approaches, with a comprehensive understanding of and appreciation for mission challenges.

For this contract we will be partnering with Halvik Corp. Halvik is a larger company of 100 employees and over 10 years of legacy experience supporting U.S. government agencies and programs. Halvik delivers a broad range of next-generation IT, and professional services to help agencies develop, maintain, and modernize IT systems, protect their networks and assets, and improve the effectiveness and efficiency of mission-critical functions for U.S. Citizens and warfighters. By partnering with Halvik, our team gains the resources and experience of a large company while maintaining the strong commitment and personalized service of a small business. A subcontract management plan will be utilized to ensure accountability. The Kaizen Halvik partnership will herafter be refered to as Kaizen and Kaizen team interchageably.

2 Software Development

2.1 Our SDLC Approach

Kaizen approach to providing software development services is rooted in our proven multifaceted methodology that includes effective management, highest level technical competence, monitoring and control of delivery schedules and milestones, documentation, adherence to industry best practice, Software Development Life Cycle (SDLC), Technology Services Life Cycle (SLC) and open standards.

Our Technology Services Life Cycle (SLC) is a set of integrated technology processes that provides process models, standards, decision guidance and tools for technology services and service delivery and is based on ISO 9001 to establish management responsibility; CMMI to establish integrated processes and procedures for handling technical inputs, manipulation, and outputs; and PMBOK for providing program and project management support and oversight.

Our processes and procedures are aligned to the 18 CMMI Key Practice Areas (KPAs) at Maturity Level 3 and are regularly routinely enhanced, updated, and internally audited. Based on our processes and procedures for those KPAs, the team uses a project management model with the following delivering high quality technical services to our clients.

CMMI 3 Key Practice Areas (KPAs)

1. Initiation
2. Project Estimation
3. Process and Product Planning
4. Data Management Planning
5. Metrics Planning
6. Quality Review Planning (Internal)
7. Validation Planning (External)
8. Risk Management Planning
9. Planning Project Tracking
10. Systems and Facilities Management
11. Project Domain and Technology Training
12. Plan for Stakeholder Involvement, Dependencies and Issues
13. Create Integrated Plan for Project Execution
14. Process Monitoring & Control (PMC) Activities
15. Configuration Management (CM) Activities
16. Management Review Meeting(MRM) Activities
17. Milestone review Activities
18. SQA Audit (SQA) Activities

Using this SLC approach for our clients, we have conducted detailed discovery efforts to determine the current state of legacy applications; supported roadmap planning and created and supervised the implementation of those plans; defined, recommended and supported the reengineering of existing business processes; documented and managed the configuration of technical artifacts; developed enterprise level systems and integrated business processes and technical services; managed the assessment, transformation and migration of large volumes of critical data, as well as supporting infrastructure redesign and build out, and supporting the transition state. Our SLC framework provides the basis for the success of our modernization efforts.

Discussed further in this section, our SLC is composed of 5 major components that help in streamlined IT delivery: **Analysis and Design**; **Software/Systems Development**; **Systems Integration**; **Infrastructure Management**; and **Data Management**.

2.1.1 Software/Systems Development

Our team has broad and proven experience supporting Waterfall, Spiral, and Agile software development methodologies. Although some organizations continue to use more traditional methodologies, Agile is rapidly being adopted in both the Commercial and the Federal software development space and is fast becoming the primary methodology. Kaizen provides Agile and SAFE Coaching support to USPTO and GPO, and has supported the migration from Waterfall to Agile for several customers. Agile leverages iterative and incremental development in which requirements, designs, and solutions are developed in close collaboration with business and IT stakeholders. Incrementally growing and testing the software results in higher quality and more timely solutions, and is a key component of significant increases in the quality of software releases and thereby reducing the total cost of implementation. Our proven Agile approach, as demonstrated in ***Figure 3***, which has been aligned to CMMI-DEV Level 3 best practices, is specifically designed to accommodate rapid and flexible responses to fluid requirements, stakeholder needs and design and development constraints.

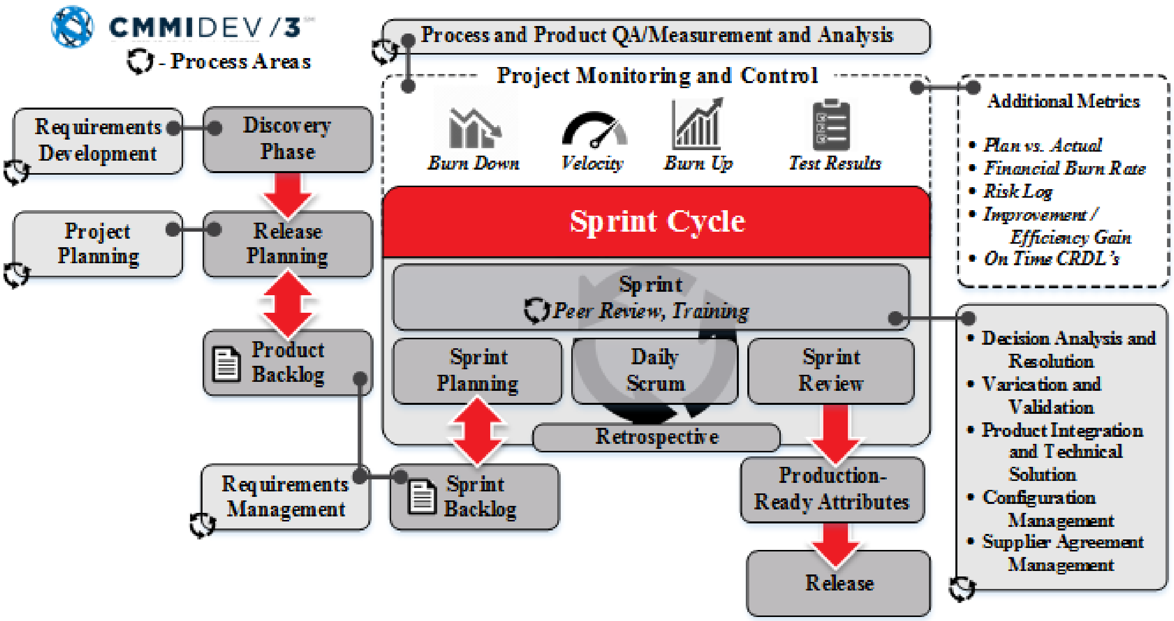

Figure 1 Our Agile Development Processes Provide Alignment with CMMI Principals to Provide enhanced Speed of Application Delivery with the Highest Quality

Our successful approach to delivering development services will provide IdeaSports with the following benefits:

- **Enable** the realization of end-to-end service functionality as quickly as possible while maintaining the ability to adjust development as priorities and stakeholder needs change;
- **Ensure** that development is aligned with the proposed solution and provides a means to address developer concerns proactively; and

- **Establish** project delivery timeframe within budget to deliver multiple end-user service capabilities for stakeholders based on priorities.

2.1.2 Systems Integration

Our team's integration practices are based on standardizing services to allow a collaborative relationship between service providers and service consumers. This standardization includes the documentation, publication, and communication of Service Definitions, Catalogues, and Web Services, based on the Service Oriented Architecture (SOA) specifications in central repositories as part of the service and component architectures. As new applications are designed, SOA based implementation practices are used to ensure that the services created to allow for a loosely coupled approach that maximizes the reuse potential by meeting the broadest set of requirements possible. Our approach in ***Figure 4*** is adaptable and customizable making it applicable to the nature of the engagement being undertaken. Our integration practice is architecture-centric, security focused and Agile in nature. Our team has considerable experience delivering large-scale integration projects that leverage our established System Integration practice. Our team has successfully employed and delivered SOA based web services integration solutions on a number of our engagements and has played a leadership role in assisting organizations to acquire and institutionalize integration technologies, frameworks and methodologies into their daily operations and practices.

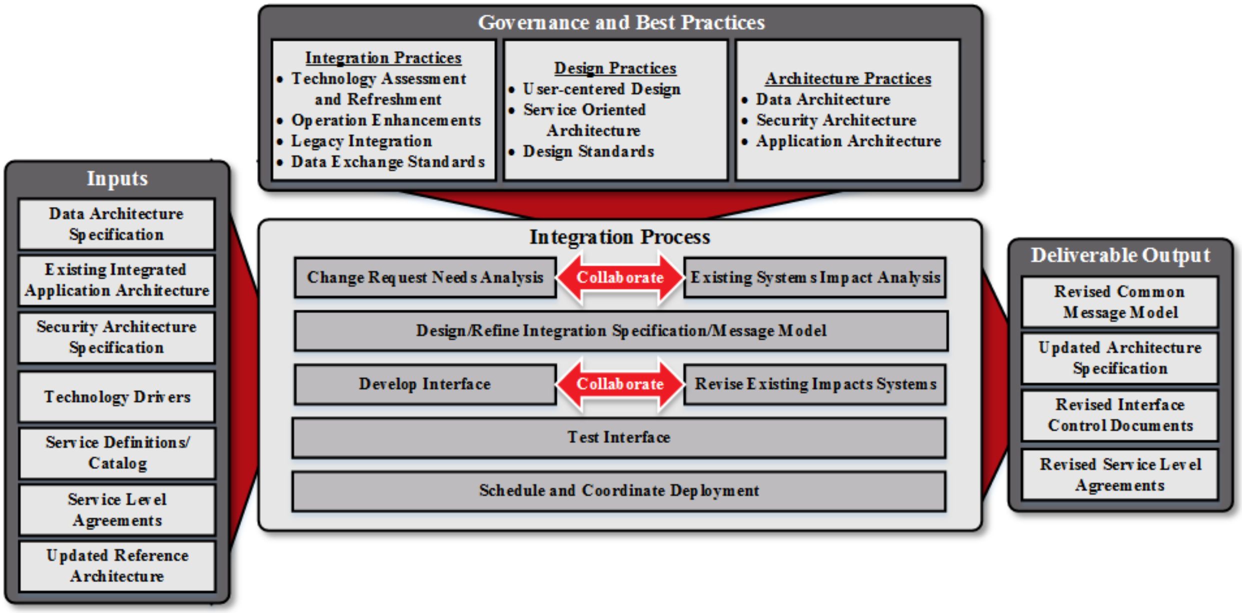
Governance and Best Practices
Integration Practices
• Technology Assessment and Refreshment
• Operation Enhancements
• Legacy Integration
• Data Exchange Standards
Design Practices
• User-centered Design
• Service Oriented Architecture
• Design Standards
Architecture Practices
• Data Architecture
• Security Architecture
• Application Architecture
Inputs
Data Architecture Specification
Existing Integrated Application Architecture
Security Architecture Specification
Technology Drivers
Service Definitions/ Catalog
Service Level Agreements
Updated Reference Architecture
Integration Process
Change Request Needs Analysis
Collaborate
Existing Systems Impact Analysis
Design/Refine Integration Specification/Message Model
Develop Interface
Collaborate
Revise Existing Impacts Systems
Test Interface
Schedule and Coordinate Deployment
Deliverable Output
Revised Common Message Model
Updated Architecture Specification
Revised Interface Control Documents
Revised Service Level Agreements

Figure 2 Our Proven Approach to Providing Loosely Coupled System Integration Platforms that Allow for Maximum Flexibility, Upgradability, and Reuse.

3 Technical Specification

This technical specification defines the high level architecture and technologies to be used in the development of the Player Portfolio Management System (PPMS). In addition, it defines the functional capabilities and high level project schedule required to develop the PPMS.

3.1 Concept of Operation

The PPMS Web Portal is the primary medium through which all users initiated interaction takes place with the PPMS. Internet users use this interface via desktop and smart mobile-devices to interact with the system. The operations performed by users vary from maintenance of personal information, view player statistics for specific game, training session, or a season or view payer specific video clips for training sessions, game, etc.

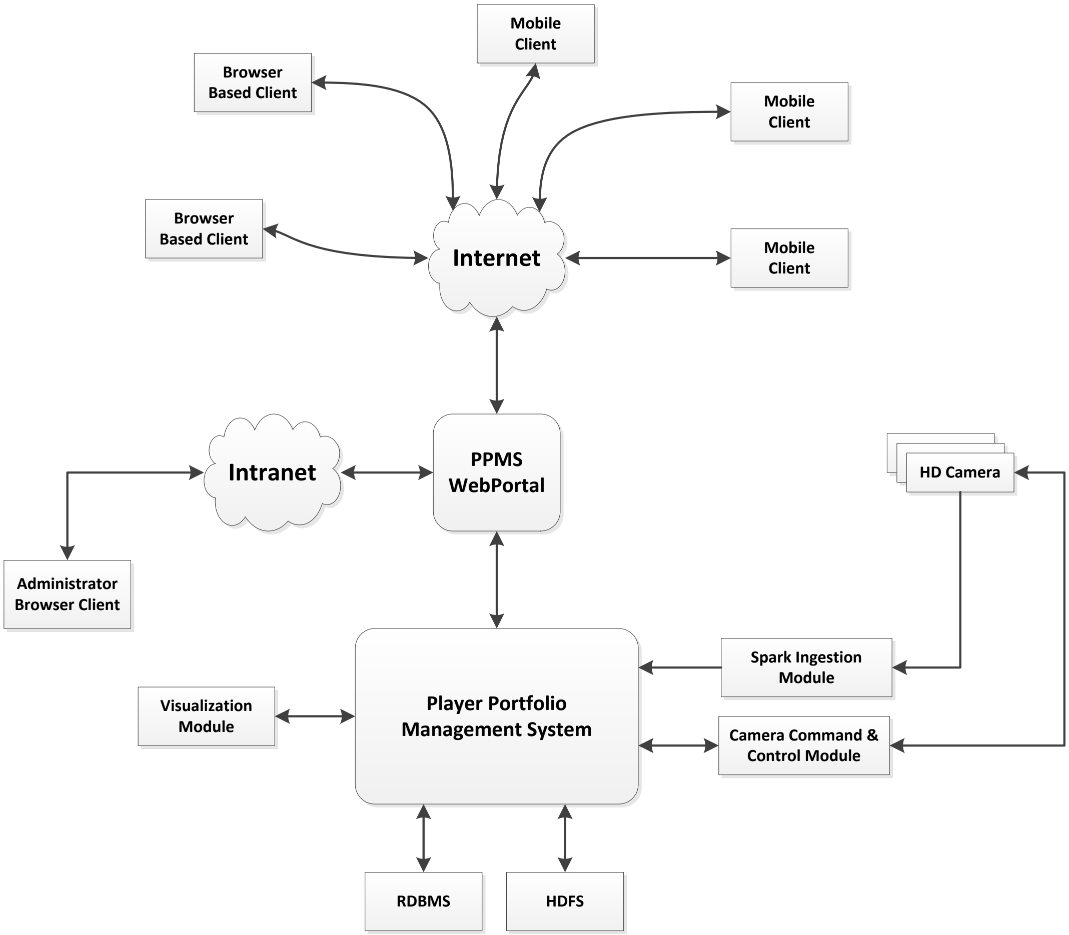

Figure 3.1: Concept of Operation

Figure 3.1 shows the high level schematics of the concept of operation of the Player Portfolio Management system. Of importance is the presentation services support via the PPMS Web Portal. All client based interaction with the PPMS is via the PPMS Web Portal services that provide support for both web browser-based clients and smart mobile-device clients. The administration client is used to configure the system inclusive of the cameras and association of the player tracking device to the cameras. This is essential for maintaining the integrity of associating a player to one or more video streams. The high definition video streams are captured via the Spark Ingestion module and store on the HDFS media. The Camera Command and Control module is used to control the cameras via the administrator client. The visualization module provide the ability to stream live and static videos at various levels of specification such as player from various angles, overall game view, replay, etc. The system will be able to captured and play back various statistical information at the player, and team levels. The system will also be able to provide predictive analytic information at various levels inclusive of individual player and team.

3.2 High Level Architecture

The following sections outline the high level architectural views of the Player Portfolio Management System. It includes components that may be deployed on various machines using a range of software technologies.

3.2.1 Layered Architecture View

The part of the software that specifically solves problems from the domain usually constitutes only a small portion of the entire system, although its importance is disproportionate to its size. To apply our best thinking, we need to be able to look at the elements of our model and see them as a system. We need to decouple the domain objects from other functions of the system, so we can avoid confusing the domain concepts with other concepts related only to software technology or losing sight of the domain altogether in the mass of the system.

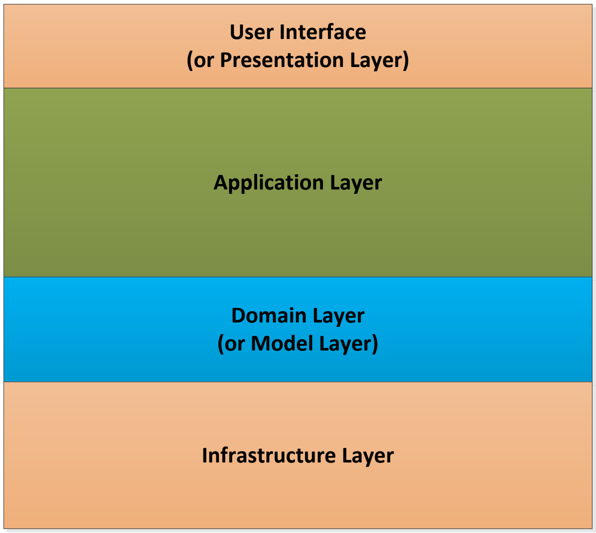

Figure 3.2: Layered Architecture View

Figure 3.2 shows a depiction of the canonical layered architecture view. There are all sorts of ways a software system might be divided, but through experience and convention, the industry has converged on Layered Architectures, and specifically a few standard layers as shown in figure 3.2. The essential principle is that any element of a layer depends only on other elements in the same layer or on elements on the layers beneath it.

The value of layers is that each specializes in a particular aspect of the software application program. This specialization allows more cohesive designs of each aspect, and it makes these designs much easier to interpret. Of course, it is vital to choose layers that isolate the most important cohesive design aspects. The most successful architectures use some version of these four conceptual layers:

User Interface (or Presentation layer)
This layer is primarily responsible for showing information to the user and interpreting the user's commands. The external actor might sometimes be another computer system or component rather than a human user.

Application Layer
The application layer defines the job the software is supposed to do and directs the expressive domain objects to work out problems. The tasks this layer is responsible for are meaningful to the business or necessary for interaction with the application layers of other software systems.

This layer is kept thin. It does not contain business rules or knowledge, but only coordinates tasks and delegates work to collaborations of domain objects in the next layer down. It does not have state reflecting the business situation, but it can have state that reflects the progress of a task for the user or the program

Domain Layer (or Model Layer)
The domain layer is responsible for representing concepts of the business, information about the business situation, and business rules. States that reflects the business situation is controlled and used here, even though the technical details of storing it are delegated to the infrastructure. This layer is the heart of the business software.

Infrastructure Layer
The infrastructure layer provides generic technical capabilities that support the higher layers; message sending for the application, persistence for the domain, drawing widgets for the UI, and so on. The infrastructure layer may also support the pattern of interactions between the four layers through an architectural framework

3.2.2 Hexagonal Architecture View

Our development methodology is based on the hexagonal architecture presented in figure 3.3 supported by a number of core Java frameworks provides a sophisticated and mature environment for the development of domain centric object oriented software applications. In our approach is focused on developing the domain model objects and then use the frameworks to address major portions of the infrastructure capabilities such as persistence and security. The hexagonal architecture pattern places the domain model in the middle, with presentation, persistence, security and other services dependent upon the domain model on the edges and accessible through ports. The ports are a mechanism for building the software scaffolding to provide the requisite service support to the domain objects.

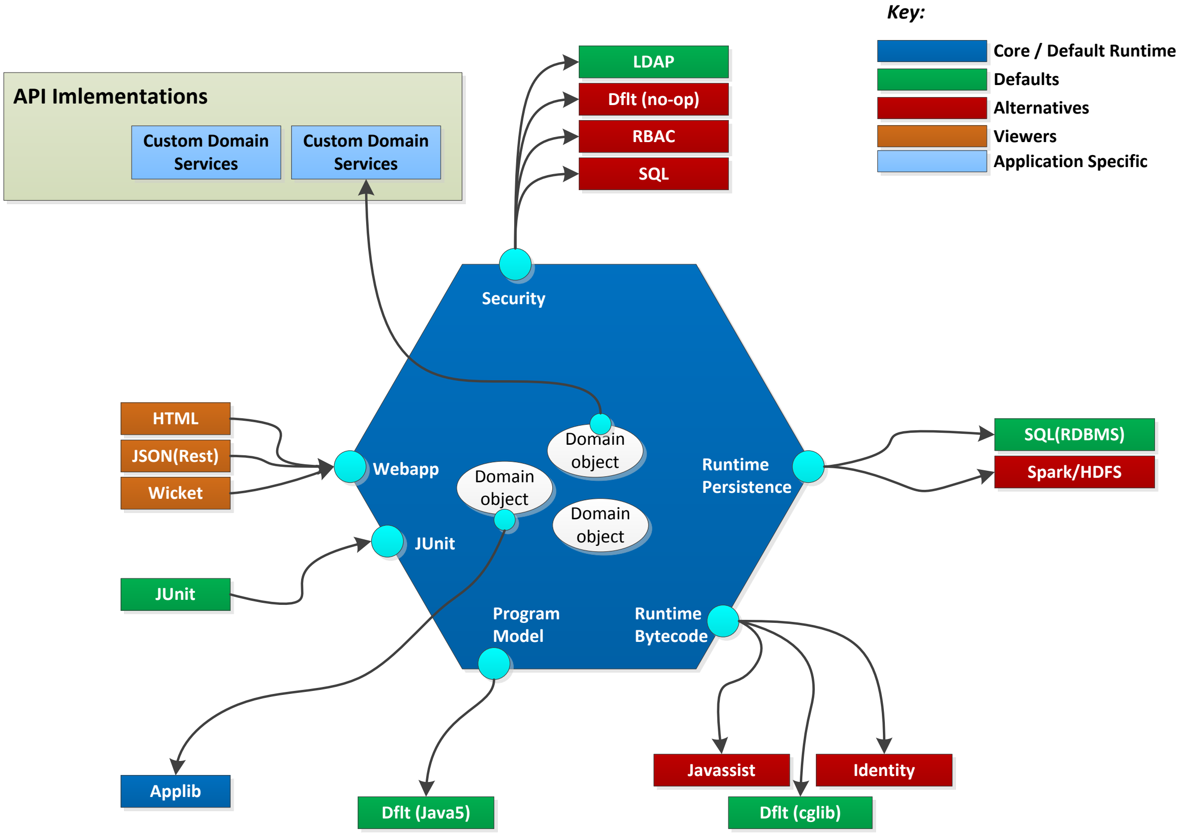

Figure 3.3: Hexagonal Architecture View

If a particular service or functional capability does not exist we would develop it along with the necessary scaffolding to support the annotation decorative mechanism. Our architecture works using convention-over-configuration where developers write POJO domain objects following a set of conventions and annotations. These are then interpreted by the Java frameworks that inject the support services in such a way that allows the domain object operation to be delegated to the appropriate services.

The core Java frameworks are extensible and customizable. We have the option of using different persistence stores, security mechanisms, and profile stores.

3.2.3 High Definition Video Ingestion

This section outlines some of the technologies we will be leveraging to capture real-time high definition video, player statistics and player vital statistics. We will be using Apache Spark framework to capture all the real-time data and to perform analysis including predictive analysis of this data. Apache Spark is a general purpose cluster computing system with high level Java API and supports a rich set of higher-level tools including Spark SQL for SQL and structured data processing, a machine learning library, a graph processing library and a streaming library.

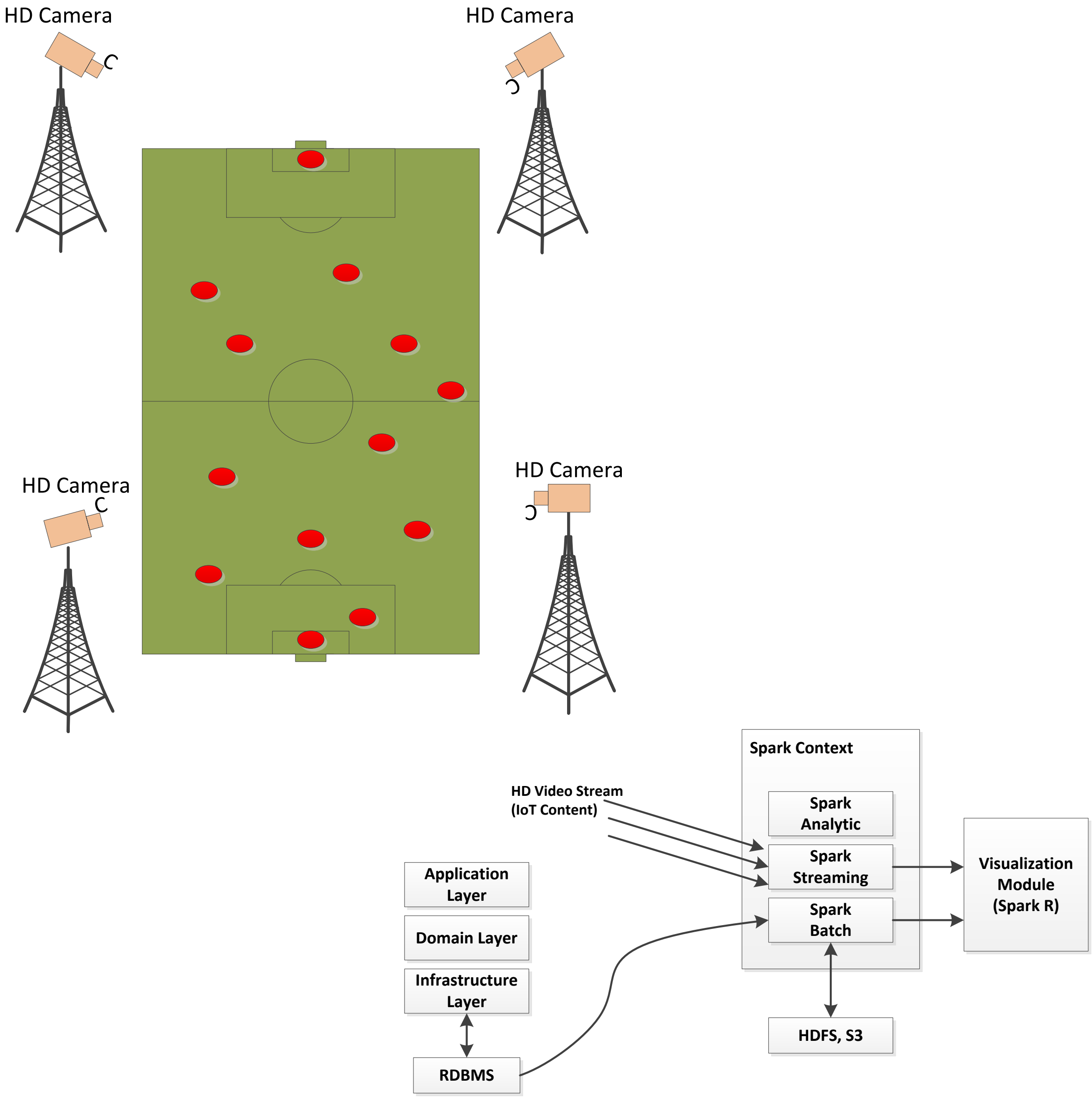

Figure 3.4: HD Video Ingestion

Spark Streaming is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams. Data can be ingested from many sources like Flume, Kinesis, or TCP sockets, and can be processed using complex algorithms expressed with high-level functions like map, reduce, join and window. Finally, processed data can be pushed out to filesystems, databases, and live dashboards. In addition, you can apply Spark's machine learning and graph processing algorithms on data streams.

4 Requirements

Kaizen shall develop, integrate, produce, test and deliver the Player Portfolio Management System with capability that meets the requirements described below

4.1 Functional Requirements

This section provides a listing and high level description of the functional capabilities that the Player Portfolio Management system will provide.

4.1.1 Create/Maintain Basic Player Information

The system shall provide the ability to create and maintain all pertinent basic information about a specific soccer player. Information will include but not limited to the following:

- Player Name
- Player Address
- Player Phone Number
- Parent/Guardian Name
- Parent/Guardian Contact Information
- Positions played
- Player Age
- Link Player to Parent

This functionality requires the creation of GUI Screens controlled by multiple roles and supported by domain objects.

4.1.2 Create/Maintain Basic Parent/Guardian Information

The system shall provide the ability to create and maintain all pertinent basic information about a specific parent/guardian. Information will include but not limited to the following:

- Parent/Guardian Name
- Parent/Guardian Address
- Parent/Guardian Contact Information

This functionality requires the creation of GUI Screens controlled by multiple roles and supported by domain objects.

4.1.3 Create/Maintain Soccer Player Statistics

The system shall provide the ability to create and maintain all pertinent soccer player statistical information. Information will include but not limited to the following:

- Miles/kilometers run
- Top Speed
- Average Speed
- Change of Direction
- Number of Passes Attempted

- Number of Passes Completed
- Passing Accuracy
- Loss of ball
- Percentage of Participations
- Movement off the ball
- Creating passing lanes
- Vital statistics
  - Heart rate
  - Blood pressure
  - Red zones/Green zones

This information will be applicable for the following scenarios:

1. Training
2. Games
3. Team – averages
4. Club – averages

Expectation is that we should be able to collect this information in an automated manner using some form of instrumentation.

4.1.4 Link HD Camera to Specific Soccer Player

The system shall be able to link a camera to a specific soccer player.

4.1.5 Camera Track linked Player

The system shall be able to continuously track and provide a video feed of a linked player.

Record the Video Feed for a Tracked Player
The system shall be able to continuously record a player that is being tracked by a specific camera.

4.1.6 Predictive Analysis

The system shall be able to perform predictive analysis. TBD

4.1.7 Users Live Stream Specific Player

The systems shall provide the ability for Internet users to live stream HD video stream for specific soccer player

4.1.8 Production

4.2 Non Functional Requirements

4.2.1 Response Time

The response time for typical operations in the Player Portfolio Management system will be consistent or better than the response time associated with comparable applications.

5 System Design

5.1 Application Language and Tool Selection

The system will be written in Java and be compliant with the latest version of the Java JRE. We intend to leverage some of the mature Java based technologies for building enterprise information systems. These technologies are primarily open source Jakarta and Apache projects. Some of these technologies include but not limited to the following:

- Java 8 JDK
- Scala
- Spring Framework
- MyBatis SQL Mapper Framework
- Apache Spark Framework
- AngularJS/Bootstrap
- Antler String Template Code Generator

5.2 Database Selection

We at Kaizen have experience using all of the major relational database management system (RDBMS) on the market, inclusive of Oracle, MySQL, MS SQL, Sybase, and Informix

5.3 Security

We at Kaizen have significant experience using Single Sign-on LDAP based authentication and authorization schemes that also integrates well with the core frameworks.

6 Schedule Estimate

TBD

7 Organization and Staffing

TBD

8 Work Breakdown Structure (WBS)

TBD