Rationale_System_Architectures_Used

The following include rationales for the system architectures used for Payday.

Client Server Architecture

- Payday's client server architecture includes a client side and a server side where the client makes requests to the server, which carries out logic to manipulate the database. The rationale for using a client server architecture for Payday is that it allows multiple clients to be active across different sessions, it allows data to be stored within a centralized database location on the server, it allows the same data to be accessed by multiple clients, it allows support for multiple platforms, there is familiarity of the architecture by our developers, and it allows flexibility for building future platform agnostic clients.

Four Tier Architecture

- Payday's four tier architecture is a four tiered layered approach that allows abstraction between different system processes. The tiers include storage layer that consists of a database and a file server, an application logic layer that consists of the server side logic, a presentation server layer that consists client side logic, and a presentation client that consists of view templates. The rationale for using a four tier architecture for Payday is that it allows modularization of code, promotes good practices for security and it allows flexibility for building future platform agnostic clients. Also a four tiered layered approach allows abstractions between all system processes and promotes better maintenance.

Restful JSON Endpoints

- Payday's application programming interface includes endpoints serving data from the database as Restful JSON from the server side of the client server architecture. The rationale for using a Restful JSON application programming interface for Payday is that is allows simple parsing by frontend JavaScript, it is extremely fast, it has an object oriented data structure, and there is familiarity of the architecture by our developers.