PassagePoint v10

Architecture White Paper

Java versus .Net

PassagePoint v10 family of products uses Java for some key advantages it provides specific to Visitor Management. The Microsoft .Net framework is inherently tied to the Microsoft Windows platform while Java is cross platform. While .Net based products will only run on Windows, PassagePoint can be run on Mac OS and Linux as well as Windows. Java also provides cross browser integration allowing applications to be deployed via standard web access methods which are inherently unavailable with .Net.

Client/Server Architecture

The PassagePoint platform is entirely client/server based. This environment provides great advantages in overall application security since any integration with external systems and databases is centrally controlled through the PassagePoint Server. This allows client stations, which tend to be in fairly insecure locations, to be strongly secured on the network as they would not need access to any other resource other than to PassagePoint Server via a proprietary client/server protocol. All external systems, such as databases, directories, and access control systems, are protected by PassagePoint Server which provides a uniform set of access controls to easily and cleanly control access of core resources by external clients.

The PassagePoint client/server architecture also greatly simplifies overall application setup and deployment. While other applications require that every client station be individually configured to have access to centralized resources such as databases, directories, and access control systems, in PassagePoint these are all centrally configured on the server and clients simply access these services through PassagePoint Server. This not only eliminates many client unique setup steps, but also results in a much more secure deployment.

Network Transparency

At the core of the PassagePoint system is a network-transparent data transport. This transport packetizes all forms of data used within the application in such a way that multiple streams of data can be sent concurrently over the same network connection. This reduces overall network load and greatly reduces network latency.

Built upon this transport is the distributed-object cache which forms the backbone of the PassagePoint application. This unique cache allows data of all types to be accessed transparently across the network regardless of the origin of the data. This means that client stations can stream video and images between them to allow network transparent device sharing, as well as allowing seamless transaction-based access to data stored in different storage systems (disk vs. relational database for example).

In addition to inherent transactional support for all data modification regardless of type (images, xml data, relational database tables, etc...), the distributed cache also uses a delta model to only transmit data that has changed during a transaction. This allows an entire application state to be rolled back in case of errors, as well as minimizing network utilization since only changed data needs to be sent to the server for storage.

Database Portability

As part of the network transparent distributed object cache, the PassagePoint architecture provides an object-relation mapping system so business logic can interact with logical objects instead of directly accessing database tables and records. This layer also provides a wrapper around all logical search methods allowing business logic to formulate database queries using relationships between business logic objects without needing to know how they are actually stored in the database. Because of this object mapping layer, all database access is tightly controlled by PassagePoint Server through a well defined set of interfaces. This makes it very easy to port this interface to different databases allowing the PassagePoint application to be entirely database agnostic.

Localization

PassagePoint incorporates extensive localization infrastructure above and beyond that provided by the underlying operating system and development toolkits. This infrastructure allows PassagePoint to re-localize the entire application interface dynamically. Since PassagePoint supports user-based language preferences, the user-interface is translates to a different user-preferred language upon login by the localization engine without needing to restart the application. This is a great convenience for regions that have more than one dominant language as both manned and unmanned stations allow the user to select the language that best suits their needs.

Hardware Abstraction

PassagePoint interfaces to a large amount of external hardware such as cameras, document scanners, signature pads, biometric devices and printers. To facilitate the addition of new hardware to the platform, the core PassagePoint architecture includes a hardware abstraction layer. This layer provides a standard interface for each type of device. To add support for a new category of device, it's as simple as implementing a low-level interface using the device-specific SDK or tools. Once the real device interface is implemented, a logical representation of each physical device is created with at the client station to

access the functionality of the real device. Combined with the network-transparent transport, it is possible for the logical device to exist anywhere on the network.

In addition, since the settings and functionality of the device is captured in the logical device definition, this means that new devices typically do not require any changes in the core application code as these devices will act and function with the same core functionality provided by other devices of the same type. This greatly reduces platform dependencies on third-party hardware implementations.

Using this network-transparent device-access infrastructure, PassagePoint allows zero configuration setup of printers. Rather than having to setup remote printer sharing on every client station, a single printer can be configured for an area through the server and other stations can share this printer using the PassagePoint device infrastructure. This allows client stations to be replaced without needing to reconfigure network printer settings.

Separation of Core and Business Logic

PassagePoint utilizes a unique two-tier architecture to combine the power and performance of Java with the ease of use of JavaScript. The core infrastructure of PassagePoint is entirely written in Java but is also completely independent of specific business logic. This core includes all low-level functionality such as networking, object-relational database mapping, security, and so on.

Built on top of this core foundation resides a JavaScript engine which provides access to every piece of core functionality via simple scripting. This layer, which is a fraction of the size of the core, implements all business logic for PassagePoint. This allows the same core to be used for multiple vertical applications, increasing overall product stability and portability. Furthermore, since all business logic is contained in replaceable JavaScript components, this opens the door for customizability of the product without requiring product rebuilds.