Data Integration

Brief Description

An IdM system is designed to be an authoritative central hub of identity information. External services may access information through APIs or directory services, or data may be provisioned to external services. It is crucial to ensure that information security is maintained when data is in transport and when stored in a new location. Changes in the IdM system should be propagated to external systems in a timely manner. The ease and speed of propagating changes may be a factor when procuring systems which need to be integrated with the IdM system.

Generic Functional Requirements

1. Information about a user should include attributes as specified by the organization
2. Only the IdM system should be able to write to log/audit data stores
3. The IdM system must be able to associate user account data across multiple systems each which may have different schemes for local identifiers
4. The IdM system needs to notify downstream systems of user-related events in a timely and secure fashion
5. The IdM system must consume upstream user-related events from systems of record in a timely and secure fashion
6. IdM functions may need to be invoked by remote systems using APIs for specific purposes

Standards Support and Integration Considerations

Where possible, avoid non-standard technologies which require specifically integrated vendor components to be deployed.

Key Design Considerations

1. Look for designs of data integration components which are loosely coupled. Components which are loosely-coupled can bring flexibility and interoperability with products from different vendors.
2. Favor designs which use commodity message queuing products. For example, use products such as Apache ActiveMQ for messaging needs.
3. Integration with downstream systems ideally should be asynchronous and loosely-coupled. For example, user provisioning can use event notification mechanisms with generic user account add/modify/delete event messages.
4. Favor systems which expose IdM system functions as REST-based services for simplicity. REST-based services allow such related systems as user administration or resource management applications to simply access IdM functions.

Technical Solutions

1. Commodity messaging products such as Apache ActiveMQ
2. Integration technologies such as Apache Camel
3. REST-based web services for API exposure to external applications
4. Vendor data integration products such as Tivoli Directory Integrator