Use Cases for extending Legacy Applications to participate in the Web Architecture

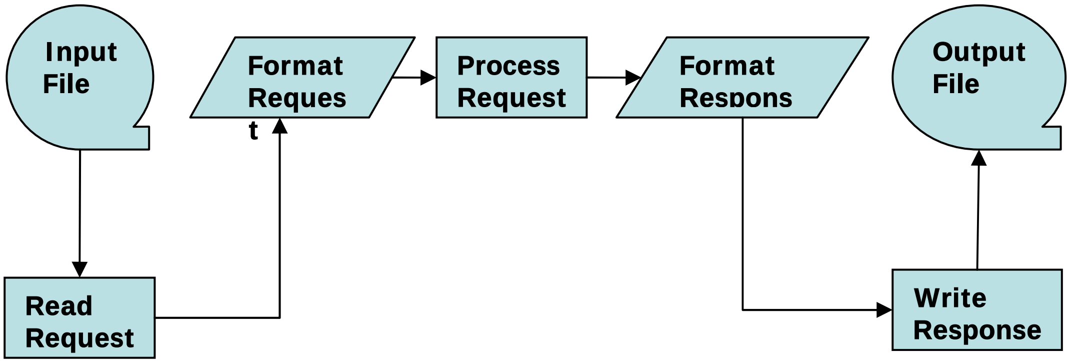
Background

In general Legacy Applications that run on IBM mainframes can be considered to fall into three archetypes:

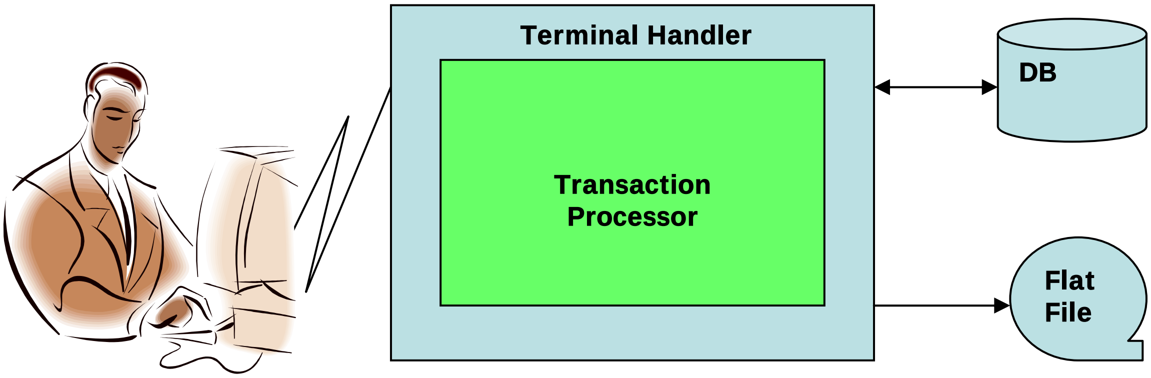
1. Batch flat file input/output applications
2. Terminal input/output applications
3. Queue based applications

In addition, the archetypical terminal and queue based applications may operate upon core business data stores in real time or they may be batch transaction collectors where input data fields are validated and batch flat file input preparation is performed.

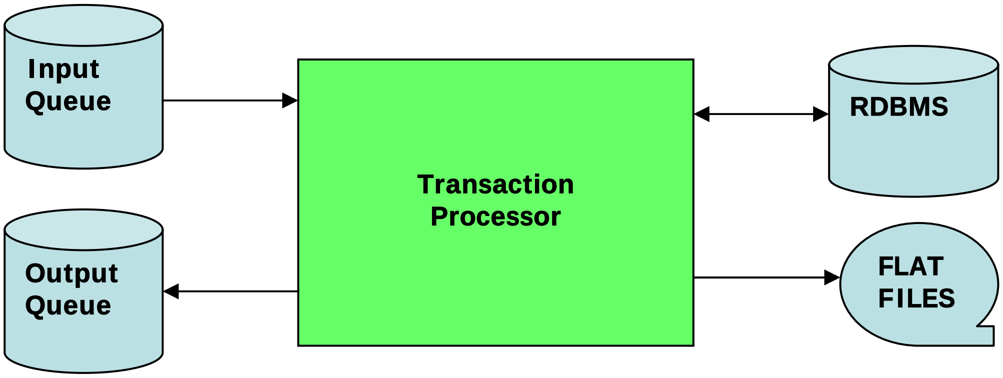
Archetype: Batch application



Archetype: Terminal based application



Archetype: Queue based application



The two archetype candidates for inclusion in Web architecture are the terminal based applications and queue based application.

USE CASE 1:

With the terminal based application several factors have to be addressed:

- **Fixed terminal screen field mapping to XML**
- **Freeform terminal screen field mapping to XML**
- **Terminal Function key mapping to action semantics in XML**
- **Security context mapping between the invocation process or end user and the inherent terminal or transaction based security profile**
- **Session inheritance**

Freeform screen displays were early attempts to create self defined "string" data streams. The challenge facing the web services architecture is the definition of an interface parser, which is able to dynamically parse legacy system data fields with the hierarchical service interface description currently used. This use case suggests that the current interface description become dynamic and utilize a message dictionary to parse data from the source system into a hierarchy and from a hierarchy into a string stream.

The security context mapping can be described as an invocation filter service that actively restricts data and actions based upon the access profile of the invoking end user without effecting code changes to the legacy applications.

Session inheritance must be a connective service that maintains application identity, state information and terminal identity between invocations of the terminal based application.

USE CASE 2:

With queue based applications the major factors needing to be addressed are the following:

- **Data type transformation**
- **Data Stream mapping to XML**
- **Security context mapping between the invocation process or end user and the inherent message or queue based security profile**

IBM mainframe applications that use queues in many cases contain special data formats found in systems based upon the MVS operating system. These special cases are bit mapped fields, packed hexadecimal numbers where every nybble represents a decimal digit 0-9, Packed decimal fields which are varied in length but every nybble represents a decimal digit and the last nybble represents the numbers sign (positive or negative). The Service interface description must contain the ability to natively transform these specialized formats into XML.

Data stream mapping has the same requirement as that mentioned above for freeform screen mapping.

Security mapping must take into account not only the security profile of the message queue and payload messages but also any sub functions described within the message that have specialized security rules. This filtering layer must have access to a similar data stream parser as that utilized for free form data streams.

Conclusion

While the batch job archetype does not lend itself to a Web Architecture, the committee could consider establishing design guidelines for creating straight through batch applications that move away from a file based input/output to a queue based input/output application.

The other archetypes are adaptable to the web but come with the challenges described above. While the IT vendor community is addressing these challenges with specialized product extensions, it would be helpful for customers to have a reference implementation pattern to hold the vendors accountable to.