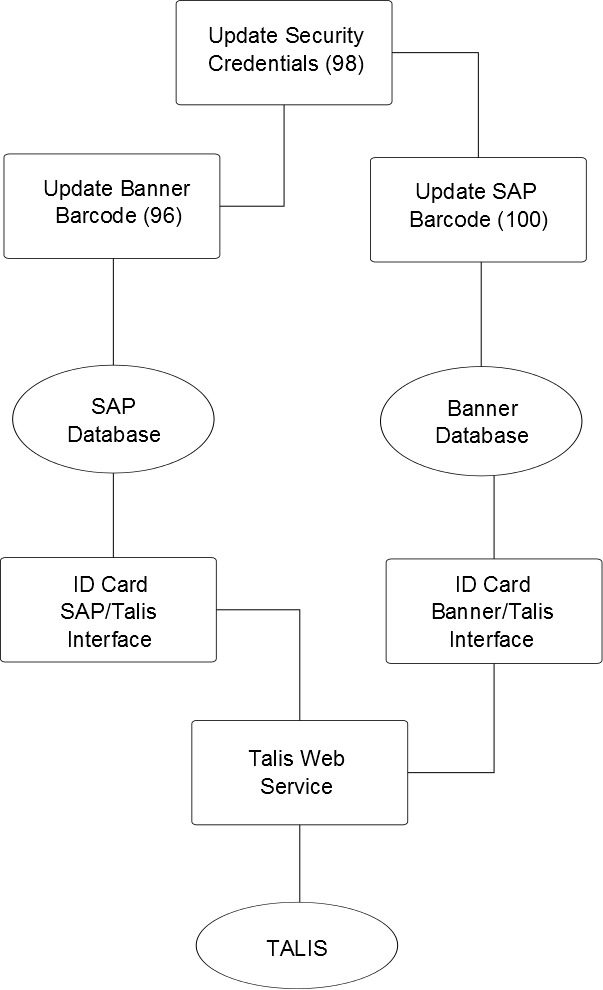
Talis Interface Replacement Design

# Requirement

Talis must be updated to hold the Barcodes assigned to a user so that the user can be granted access to the Library systems and any downstream systems dependent on it.

# Existing System



The existing system starts the process of updating Talis when the Update Security Credentials process runs on AccMan. Process 98 queries Janus to retrieve the newly assigned card details for users on a periodic basis, and when it finds a new set of details it will create transactions and/or perform actions according to the user’s type. In the case of a student, one of the transactions it creates will trigger the 96 process (Update Banner Barcode), whereas for staff and associates it will be 100 (Update SAP Barcode).

These processes update a database table on the relevant system (ZCARDHIST for SAP, and T\_IDCARD\_DATA for Banner).

The next step in the process is performed by the Perl ID Card Talis bridge code. This queries the tables, retrieves the requisite information, and calls a web service running on the Talis server (UpdateBarcode) to update the user’s information on Talis with the newly assigned barcode.

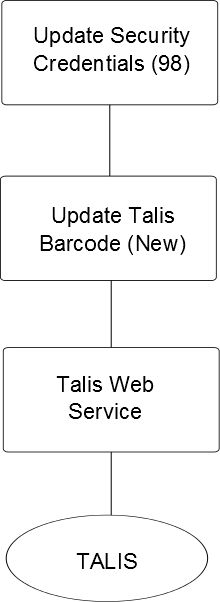
# Reason For Replacement

The current SAP system is being updated, and during the update process a table was identified (ZCARDHST) as not being part of the core SAP database. Investigation showed that it was being used as part of the interface that transfers barcodes into Talis from the upstream systems as part of the AccMan process flow when new staff members are assigned an ID card.

This table will not be present once the SAP upgrade is in place, and as a result the existing code will need to be altered as a matter of urgency. The code is currently written in Perl is not written in such a way as to make changing databases (or database locations) a trivial task, so it will be a simpler and quicker approach to replace the existing code with a .NET version that would perform the same tasks and be aligned to current development standards. As an added bonus, it would allow us to combine the Banner version of this codebase (used for students instead of staff) with the SAP codebase, thus eliminating two Perl processes and combining two existing .NET processes (for updating the SAP and Banner database tables with the ID card information) into a single process.

The ALMA system will be replacing TALIS at some stage. It is anticipated that this will take place within the next 12 months. As such, this solution is only a temporary stop-gap until the new ALMA system is in place, and ALMA will itself require some further rewrites. However, the SAP system will be replaced in very short order, and ALMA will not be in place at this time, so this update is required to continue normal university function.

# Proposed Replacement



The proposed replacement is a simplification of the existing system to a single stream. A new ‘Update Talis Barcode’ process is created with the sole purpose of calling the Talis web service with the required details.

The following code amendments are required to make this possible:

1. Amend the Update Security Credentials (98) process to remove references to processes 96 and 98, and replace with the new Update Talis Barcode process.
2. Create the new Update Talis Barcode Process and associated configuration artefacts.
3. Create the Talis DAO class (and associated interface class) required to contact the Talis Web Service.
4. Potentially, a further class may be required to contact the web service depending on how this service must be contacted.
5. The web service call as contained in the Perl script is a direct http request using parameters. This is obviously insecure (as the parameters are passed in cleartext). Also, one of the parameters is a ‘transaction\_id’ parameters, which appears to be sequential but ultimately controlled by the existing Perl bridge code. Some further investigation will be required regarding the nature of this parameter, and potentially the generation of this ‘transaction\_id’ parameter may require some additional coding to encapsulate any additional complexity. It is hoped, however, that the value of ‘transaction\_id’ can be the same as that of the Transaction generated in the database to perform this action.

# Development Timescales

The amount of work involved should be in the region of 2-3 days depending on the complexity of requirements around the ‘transaction\_id’ as detailed above. In addition, there will be a requirement to perform some degree of regression testing on Process 98 (Update Security Credentials) to ensure that the requisite transactions are generated, as well as downstream testing of the Talis Web Service call itself.

## Potential Issues

1. It is unknown whether a development Talis system is available for testing the web service.
2. The ‘transaction\_id’ parameter needs to be investigated further as detailed above.
3. The insecure call to the web service may need to be addressed. This is beyond development scope, however.