UNIVERSITEIT·STELLENBOSCH·UNIVERSITY
jou kennisvennoot • your knowledge partner

Stellenbosch University Software Standards for External Developers

The purpose of this document is to communicate standards that will be acceptable to the University's IT Division when adjudicating proposals and tenders by external (non-IT Division) developers for software applications and systems for the University.

Proposals and tenders that do not meet with the standards as set out below will not be considered.

The standard is a dynamic document so that a IT's capabilities and skill expand in a sustainable manner, so the standard can be extended, for example to include more software languages.

1. **Security**
   a. Authentication
      i. Applications that require authentication for members of SU's community must use either of the following methods:
         a. SU's Single Sign-on (SSO) server
         b. SU's central LDAP directory
      ii. Under no circumstances may SU passwords be intercepted and stored in an application/system's database.
   b. Encryption
      Where confidential or private information will be exchanged between the application/system and the user's client and/or an SU institutional information system, SSL encryption must be provided.
   c. Authorisation
      1. The application/system must implement role-based access control. Using multiple usernames to distinguish between roles and embedding roles into usernames are not acceptable.
      2. If the standard roles (e.g. student, academic staff, alumnus, etc.) as supplied by SU's Identity Management (IDM) System are not relevant or appropriate, roles and privileges must be managed within the application/system.
      3. SU's "line management hierarchy" as defined in its HRMS system, must be used in workflow and approval processes, where relevant.
   d. Standard ports
      Only standard ports will be opened for communications through the SU's firewall.
   e. Audit trails
      The system/application must be capable of generating and maintaining an audit trail of logons and transactions that identify the user or system that took specific actions and the date and time of such actions.

2. **Integration**
   a. The application/system should strive to fit into SU's Service Oriented Architecture (SOA) methodology. It should be able to consume and produce web services (the SU is implementing a set of common functions against institutional information as web services) and componentise functions.
   b. The application/system must be able to integrate on the application and data levels with SU institutional systems using SU's near-real-time XML advanced queue messaging mechanism and web services.
   c. Various standard database adapters are available, but will only be considered in exceptional cases. In general, we will not allow direct integration with institutional databases.
   d. A browser-based user interface (UI) is preferred. Thick clients will only be considered under exceptional circumstances. All processing must occur server side.

UNIVERSITEIT·STELLENBOSCH·UNIVERSITY
jou kennisvennoot • your knowledge partner

    e. Where relevant (in most cases where the target users are staff or students), the application's UI must be integrated into SU's institutional web portals. Portlets (or web parts) and services should comply with JSR168 and WSRP standards.

    f. Lightweight integration protocols for "mashups", such as REST, will also be considered.

    g. The integration interfaces must cater for error trapping, monitoring and audit trails.

3. Platforms

    a. Only J2EE or .NET platforms will be acceptable.

    b. Systems/applications must implement or use Oracle or MS SQL databases.

4. Programming languages / frameworks

    a. The following languages are acceptable:
        i. Java
        ii. C#
        iii. C++

    b. The following web development technique is acceptable:
        i. AJAX

    c. Where web services must be orchestrated into business processes (including workflow), the application must be compatible with Oracle BPEL.

5. Performance

The application/system's performance must be tested and certified to perform at predefined levels under specified loads.

6. Monitoring

    a. It must enable or facilitate the insertion of software performance probes in order to diagnose performance bottlenecks.

    b. It must be able to report on usage of the application (e.g. number of concurrent users instantaneously and over time, the nature of usage, etc.)

    c. Web-based applications must be capable of including SU's web analytics code on each page.

7. Browsers

Web-based applications must support Internet Explorer and Firefox up to the current version minus 2 major versions.

8. Support and maintenance

    a. The proposal must include a plan and budget for knowledge transfer to SU IT Division staff.

    b. Where the application/system is a bespoke development for SU, source code must be provided.

    c. All applicable technical documentation must be provided.

End.