

# **NCDPI**

## **Technical Standards**

Author(s): Technical Architecture Team

Last Revised: October-16, 2006

Version: 1.5

Status: Final Draft

Access: public

## REVISION HISTORY

Rev. #	Revision Date	Revised By	Description	Filename
1.0	May 23, 2006	Mike V	Initial Draft	DPI-Technical Standards.doc
1.1	Oct 4, 2006	Mike V	Further enhancements	DPI-Technical Standards.doc
1.2	Oct 5, 2006	Joe Dietzel	Review	DPI-Technical Standards.doc
1.3	Oct 5, 2006	Joe Dietzel	Review	DPI-Technical Standards.doc
1.4	Oct 16, 2006	Steve McCutchin	Review	DPI-Technical Standards.doc
1.5	Oct 16, 2006	Mike V	Released as final draft	DPI-Technical Standards.doc

## Table of Content

<b>REVISION HISTORY .....</b>	<b>2</b>
<b>1. HIGH-LEVEL STANDARDS.....</b>	<b>7</b>
<b>2. DATABASE STANDARDS.....</b>	<b>8</b>
2.1 DATABASE-ENGINE .....	8
2.2 DATABASE-MODELING TOOLS .....	8
2.3 DATABASE-QUERY TOOLS .....	8
<b>3. DATA STANDARDS .....</b>	<b>9</b>
3.1 INTERCHANGE OF INFORMATION.....	9
3.2 STUDENT DATA (UNIQUE IDENTIFIERS) .....	9
<b>4. APPLICATION STANDARDS .....</b>	<b>10</b>
4.1 AUDIT-TRAILING / LOGGING.....	10
4.2 BROWSERS .....	10
<b>5. WEB-SERVER STANDARDS.....</b>	<b>11</b>
5.1 WEB SERVER.....	11
<b>6. APPLICATION SERVER STANDARDS .....</b>	<b>12</b>
6.1 APPLICATION SERVER.....	12
6.2 COMMUNICATION STANDARDS .....	12
<b>7. SECURITY STANDARDS.....</b>	<b>13</b>
7.1 AUTHENTICATION SERVICE .....	13
7.2 WEBPAGE-ONLY SECURITY .....	13
7.3 DATA CONFIDENTIALITY .....	13
7.4 DATA TRANSMISSION SECURITY.....	13
<b>8. PROGRAMMING STANDARDS .....</b>	<b>14</b>
8.1 PROGRAMMING LANGUAGE .....	14
8.2 CODING STANDARDS .....	14
8.3 EXTRACTION-TRANSFORMATION-LOAD .....	14
8.4 TESTING TOOL STANDARDS.....	14
<b>9. REPORTING STANDARDS .....</b>	<b>15</b>
9.1 REPORTING TOOL .....	15
9.2 ANALYTICAL/DATA-MINING TOOL .....	15
<b>10. PLATFORM STANDARDS.....</b>	<b>16</b>
10.1 SERVER OPERATING SYSTEM .....	16
10.2 CLIENT OPERATING SYSTEM.....	16
<b>11. CUSTOM-CODING STANDARDS .....</b>	<b>17</b>
11.1 USE OF APIS .....	17

---

11.2	PROGRAMMING LANGUAGE.....	17
11.3	OBJECT-TO-RELATION MAPPING.....	17
11.4	REPOSITORY STANDARD.....	17
11.5	DEVELOPMENT TOOL.....	17
<b>12.</b>	<b>GRAPHICAL-USER-INTERFACE STANDARDS .....</b>	<b>18</b>
12.1	ADA/508.....	18
<b>13.</b>	<b>TECHNICAL ARCHITECTURE STANDARDS.....</b>	<b>19</b>
<b>14.</b>	<b>DOCUMENTATION STANDARDS.....</b>	<b>21</b>
14.1	DATA DICTIONARY .....	21
14.2	APIs .....	21
<b>15.</b>	<b>APPENDIX .....</b>	<b>22</b>
15.1	REFERENCES TO OTHER DOCUMENTS/STANDARDS .....	22

## **Purpose of this Document**

This document describes the technical standards used by NCDPI. These standards are currently under review and may require further analysis specifically for integrating current applications.



## 1. High-Level Standards

On a summary level NCDPI standards are based on fundamental principles. Such principles should apply to any new implementation and hopefully extend to applications being migrated from older architectures. The key principles are:

- Buy vs. Build Analysis: Buying a solution and working with the vendor on enhancements and maintenance is a better fit for NCDPI than custom-build and very expensive Software-Development-Lifecycles.
- Open Source products have greatly improved in the market spaces and do provide quality and stability. Open Source is free-of-charge software with a high level of functionality and using up-to-date technical architecture and implementation.
- Applications must be database-based: To be able to interface with the application using standards a database should always be used by the application. The ability to retrieve and interface with the data is key to a success of an enterprise application.
- Application must be web-based: To keep software maintenance and client maintenance on a manageable level the application must work using a client browser. Fat or thin client/server applications can no longer be considered in the IT landscape.
- Service instead of application hosting: Implementing a business solution using a service rather than hosting the application ourselves has significant advantages.
- Versioning: NCDPI desires to maintain the IT business solutions by using the most current version or most current version minus one (Major Release). Example: Oracle 9.x is currently supported and an accepted standard, but Oracle 10g is the most current version.

## **2. Database Standards**

### ***2.1 Database-Engine***

NCDPI is using the statewide ITS-Oracle contract for implementing database services. The contract was put in place based on the licensing requirements of NCWISE and should be extended to all other applications.

The current acceptable version is Oracle 10g Release 2. For enterprise-level applications consider using an Oracle-RAC (Real-Application-Cluster) implementation to enhance the environment with scalability, fail-over, and clustering capabilities.

### ***2.2 Database-Modeling tools***

NCDPI is using ERWIN as the tool of choice for database modeling activities. Tools, which are part of the standard Oracle licensing package, are also considered to be a standard for the department. ERWIN is primarily used by database administrators and data-modelers that greatly rely on Entity-Relationship-Modeling, analysis of different versions of schemas, etc.

### ***2.3 Database-Query tools***

NCDPI is using TOAD as the tool of choice for database queries and overall DML activities.

## **3. Data Standards**

### ***3.1 Interchange of Information***

NCDPI highly desires to exchange data between applications via XML data format. This approach has significant advantages over the other approaches. The department is currently evaluating specific XML standards (such as SIF and others) which further specify the elements and structure of information using XML.

### ***3.2 Student Data (Unique Identifiers)***

Any application storing student-related information must have the ability to store the NCWISE unique identifier as well the SIMS identifier. A third field needs to indicate which identifier, SIMS or NCWISE, is currently used for the student<sup>1</sup>.

---

<sup>1</sup> A student could be an active NCWISE student but moved to a non-NCWISE LEA, therefore the SIMS id is the most current did to be used for that student.

## 4. Application Standards

### ***4.1 Audit-Trailing / Logging***

For debugging and incident management any application needs to provide audit trail and logging information which allows operators and functional experts to understand what application function is potentially malfunctioning. The audit trail information should include at least the following information:

- Timestamp of the log entry
- User name who is triggering the application function
- Description of the function the user or application is trying to use

The level of detail of the information must allow a subject matter expert to understand how the user (or program) is interacting with the application.

For performance and security reasons the audit trail and logging should be configurable and have the option to suppress/disable the audit trail/logging features. Logs & audit data must also be stored so as not to be easily tampered with and rotated on a regular basis (all configurable).

### ***4.2 Browsers***

Browser-based application interaction is one of the key goals of NCDPI which allows the department to establish a simple and clear standard. Over the past years multiple browsers have been developed and the department started to see compatibilities issues. The currently supported browser standard is:

- Internet Explorer 5.x or greater (Windows clients)
- Mozilla/Firefox greater than 1.0 (Windows, Linux, Mac-clients), preferably 1.5.x
- The use of the Mac-based Safari browser is currently under review.

## **5. Web-Server Standards**

### ***5.1 Web server***

NCDPI prefers the use of a Apache/Tomcat server. This server supports and promotes the use of a physical, 3-tier application implementation. Products such as Weblogic, Websphere, and Oracle support the implementation of Web and application servers on the same physical hardware. Under no circumstances should the application architecture be based on the execution of business logic code within the web-tier. Therefore, additional licensing maybe required in case of Weblogic and Websphere. This can be addressed by the open-source, free-of-charge web server from Apache.

## **6. Application Server Standards**

### ***6.1 Application Server***

NCDPI's preference is the use of Weblogic, Websphere. Weblogic being the most desirable application server. BEA's product is the most stable and reliable application server. NCDPI has encountered great difficulties with the Oracle application server in the past and is therefore avoiding this application server.

The technical architecture can vary from installation to installation but should always be based on a 3-physical-tier implementation to be able to address clustering, fail-over, and scalability across all tiers of the application.

For smaller lab-based installations such as feasibility studies, open-source application servers such as JBOSS can be used.

The technical team is currently discussing the impact of using JBOSS on the enterprise level.

### ***6.2 Communication Standards***

Application services should be exposed as Web-services (SOAP, WISDL) which are de-facto standards within the industry. Such interfaces allow a very flexible approach towards application integration and should be based on the most current version or current version minus one.

These communication standards are selected to be the base of the application integration.

## **7. Security Standards**

### ***7.1 Authentication Service***

NCID is the mandatory authentication service for NCDPI. This service is hosted by ITS and is based on an Oblix (now Oracle) implementation of an LDAP-based service (Core ID).

### ***7.2 Webpage-only security***

Webgate is the agency's standard protection mechanism for protecting web based applications. , This service is fully integrated into the NCID system and authenticates the user (using authentication browser cookies) into web application and can be used for both Internet Explorer and Apache web servers.

### ***7.3 Data confidentiality***

Confidential data such as Social Security Numbers, if required, should be stored in an encrypted format (at least hashed) to ensure that operational personnel do have access to them. If the application cannot handle this requirement, the information should at least be masked (last 4 digits of the number are disclosed).

Reports and Graphical User interfaces should have the option, through system super user accounts, to mask information

Usernames and passwords should always be fully encrypted or hashed when stored in the database for authentication purposes.

### ***7.4 Data Transmission Security***

Any system which receives or transmits confidential information (SSN user names/passwords etc) is required to transmit that information over a secure channel such as SSL/TSL, Secure FTP, Secure copy Virtual Private Network, or to encrypt the data prior to transmission via the use of PGP or other encryption system.

## **8. Programming Standards**

### ***8.1 Programming Language***

Java is the preferred programming language standard for NCDPI. Applications using this programming language can run on almost all current environments within NCDPI.

All critical standards of Java (JRE, JDK, J2EE, JNI, JNDI, etc) are published and available as source code.

The supported version should be the most current one or the previous of that latest version.

Application programming interfaces should be provided by a Java interface which allows the developer to use Java code to communicate with the application for integration purposes.

### ***8.2 Coding Standards***

Coding standards must adhere to the standard published by SUN (<http://java.sun.com/docs/codeconv/>) from April 20, 1999.

### ***8.3 Extraction-Transformation-Load***

SAS BI provides all ETL capabilities required to extract, transform and load data from one system to another. As this document is written extensive training is being conducted by NCDPI to be able to implement, maintain and support data transfer activities between the systems.

### ***8.4 Testing Tool Standards***

Mercury is the enterprise-level testing tool selected by NCDPI. ITS is currently building a Mercury test tool service which allows multiple agencies to use this service to verify production readiness and appropriate load-related thresholds. All enterprise-level applications should be load tested to avoid performance degradation in productions. The scripts used for the load test must be written using the Mercury tool.

## **9. Reporting Standards**

### ***9.1 Reporting tool***

NCDPI uses the statewide SAS contract which provides the SAS-Business Intelligence tools.

The most current version supported by NCDPI is SAS-BI-9 and includes Weblogic as the application server and Apache/Tomcat as the web server.

### ***9.2 Analytical/Data-Mining Tool***

Please see reporting tool above.

## 10. Platform Standards

### ***10.1 Server Operating system***

NCDPI desires the following operating systems:

- Linux: All new system implementations should use Linux (Redhat-Enterprise Version) as the preferred platform. This open source operating system provides all missions critical services for an enterprise.
- Windows: Windows is the second choice for an operating system. Windows and Linux are dominant operating systems in the market but Linux has better enterprise-level services per unit of price than Windows (i.e. clustering).
- Solaris: Although not preferred, Solaris is the operating platform for many enterprise applications which have not migrated to Linux.
- AS400: NCDPI still has many applications on this platform. The strategic direction is to either freeze or migrate applications off of this platform and move them towards Linux or windows. This is a long-term goal.

### ***10.2 Client Operating system***

NCDPI desires to only support Windows XP and Mac OS10 as targeted client platforms.

## **11. Custom-Coding Standards**

### ***11.1 Use of APIs***

The solution should use as many APIs as possible for the implementation.

### ***11.2 Programming Language***

Please see above.

### ***11.3 Object-To-Relation Mapping***

In order to map relational database objects into programming objects an object-to-relation mapping API should be used to avoid unnecessary coding and therefore provide higher quality of the implementation. Java provides the hibernate-API which is used widely and strongly recommended.

### ***11.4 Repository Standard***

NCDPI uses CVS on the Linux (Redhat-Enterprise) platform as the code repository.

### ***11.5 Development Tool***

NCDPI uses Eclipse as the development of choice. The tool can be extended with add-ons to address specific API enhancements such as MyEclipseIDE which provides plug-ins for Web-services, Hibernate, Logging, etc.

The current version of Eclipse is 3.2.x.

Source code provided by vendors should be based on an Eclipse project which can be imported successfully into the developers Eclipse application.

## **12. Graphical-User-Interface Standards**

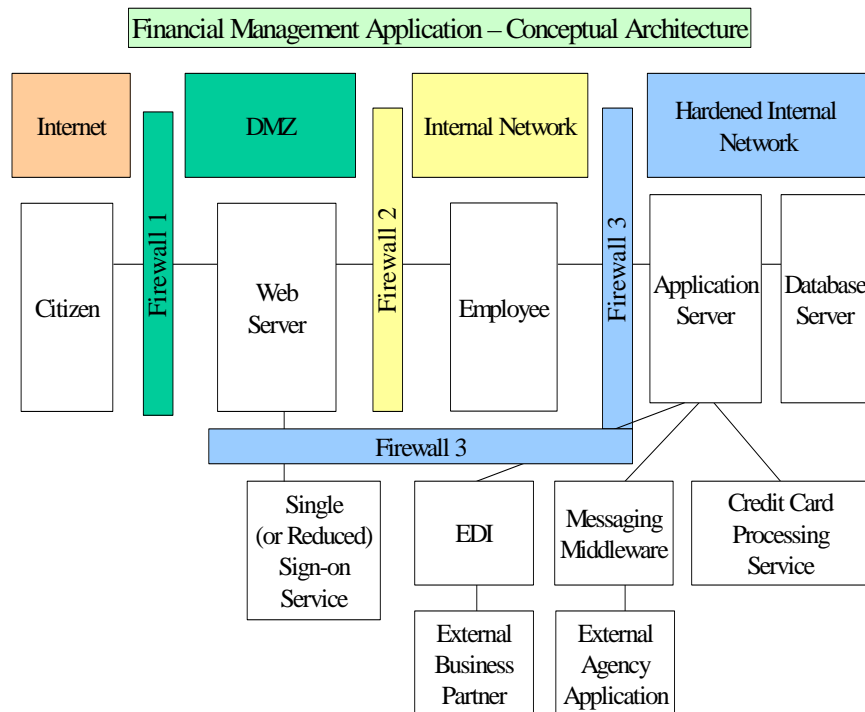
### ***12.1 ADA/508***

Use of the 508 standard when required.

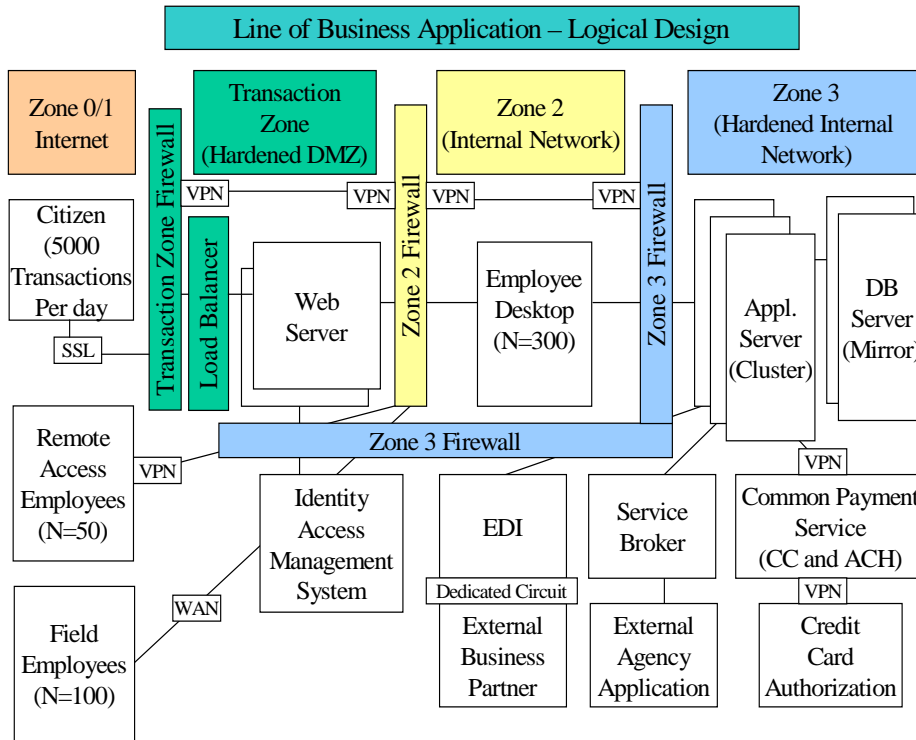
### 13. Technical Architecture Standards

The technical architecture of the solution should be based on a 3-physical tier architecture, which decouples each tier by firewalls for maximum data security and protection.

The following diagram depicts the high-level technical architecture based on an implementation example:



The following diagram depicts the access points to the tiers as well as the type of protocol used for communication (as a example).



## **14. Documentation Standards**

### ***14.1 Data Dictionary***

NCDPI requires all applications to maintain a data dictionary of all contents of the database.

### ***14.2 APIs***

NCDPI requires all applications to documentation on all supported APIs.

## 15. Appendix

### *15.1 References to other documents/standards*

Standard	Reference Document
1. Java Coding Standards	<a href="http://java.sun.com/docs/codeconv/">http://java.sun.com/docs/codeconv/</a> from April 20, 1999.
2. Statewide Technical Architecture	<a href="http://www.ncsta.gov/">http://www.ncsta.gov/</a>
3. Statewide Information Security Manual	<a href="http://www.scio.state.nc.us/SITPoliciesAndStandards/Statewide_Information_Security_Manual.a">http://www.scio.state.nc.us/SITPoliciesAndStandards/Statewide_Information_Security_Manual.a</a>
4. Security Standards and Policies	<a href="http://www.scio.state.nc.us/sitPolicies_List.asp?Other">http://www.scio.state.nc.us/sitPolicies_List.asp?Other</a> Security Standards and Policies
5. Old Policy to New Security Policy crosswalk	<a href="http://www.scio.state.nc.us/documents/docs_Active/Statewide">http://www.scio.state.nc.us/documents/docs_Active/Statewide</a> Information Security Standards and Policies Manual/crosswalk.pdf
6. Statewide IT Glossary	<a href="http://www.scio.state.nc.us/documents/docs_Active/Statewide">http://www.scio.state.nc.us/documents/docs_Active/Statewide</a> Information Security Manual/Glossary of Terms.pdf