

Best Practices for Developing Mobile Applications: The Data Driven Approach

SYBASE iANYWHERE

TABLE OF CONTENTS

- 1 Introductions
- 1 Summary
- 1 Online applications
 - 1 Advantages
 - 1 Disadvantages
- 2 Occasionally connected smart client applications
 - 2 Advantages
 - 2 Disadvantages
- 2 Best practice steps
 - 2 Step 1 – Choose your database and data synchronization infrastructure
 - 3 Why SQL Anywhere: technology considerations
 - 3 Step 2 – Design your database and synchronization infrastructure
 - 4 Synchronization design considerations
 - 4 Database design considerations
 - 5 Step 3 – Choose your application development approach
 - 5 Web applications
 - 5 Mobile-specific development tools
 - 5 Language of choice
 - 6 Step 4 – Choose your remote management approach
 - 6 Full-featured mobile device management and security
 - 6 Web application deployment
- 6 Conclusion

INTRODUCTION

In today's aggressive business environment, organizations are looking for competitive advantages in all areas of their business. Every department in an organization, including IT, is being asked to improve their processes and increase the overall competitiveness of the business. Every department must view itself as crucial to an organization's success.

Increasingly, organizations are looking to improve their productivity and competitiveness through the use of mobile computing by providing their employees with mobile devices. A well-designed mobile application can enable an organization's employees to be more productive and deliver improved customer service while reducing operating costs. However, mobile applications require innovative approaches to employing critical business applications and data. Any data or application that is used must integrate with and leverage an organization's existing application and database infrastructure.

There are two primary models for mobile applications: online applications and occasionally connected smart client applications. Online applications require a network connection to a backend enterprise data source, and are usually browser-based. Occasionally connected smart client applications are characterized by a local data store and local application, and do not require a constant network connection. Each of these models has advantages and disadvantages. It is important to understand these advantages and disadvantages so that the appropriate model can be chosen for a given business need.

SUMMARY

This iAnywhere whitepaper will explore the two mobile application models, discussing the advantages and disadvantages of each. Based on over ten years of working with customers to design, develop, and deploy mobile applications to the front lines, a series of best practice steps will be presented for developing occasionally connected smart client applications. These steps outline the lowest risk approach, increasing the probability that the mobile application project will succeed.

ONLINE APPLICATIONS

An online application does not store any data or application logic on the mobile device. Therefore, when the user requires access to the application and data, a network connection is required. Typically with mobile devices, this network connection is wireless, either WiFi or Wide Area Network (WAN). All data and applications are transmitted through this wireless network connection only when requested by the user.

Advantages

The main advantage of this model is that there is no data or application actually stored on the device, outside the organization's data center. When data is required, it is retrieved directly from the enterprise's back-end data store. Because of this, there is no software distribution to the mobile device and the user will see enhancements to the application as soon as they are made on the central server. Security is seemingly easier to manage because no data is actually stored on the mobile device.

Disadvantages

There are some significant disadvantages to the online model that primarily occur in two areas: user expectations and network communications.

A user's primary expectation is that the application and data will always be available. But with an online application, the user must have wireless network coverage. If the user is located in an area without network coverage, the application cannot be used and if the connection is slow or unstable, use of the application can be frustrating at best.

Because online applications require communication over a wireless network, they are typically much slower than the equivalent application running locally on a mobile device. Furthermore, the constant use of wireless transmissions can significantly shorten battery time and in many cases, the extensive network connection time that is required is very expensive.

OCCASIONALLY CONNECTED SMART CLIENT APPLICATIONS

An occasionally connected smart client application uses a local application together with a local database. No network connection is needed for the user to access information. The local application is designed to retrieve requested information from the local database on the device. The data in the local database is kept up to date using synchronization.

Advantages

There are many advantages to the smart client model. First of all, when a user needs data it is instantly available, without waiting for relatively slow network transmission. Furthermore, wireless network connectivity is not required to access the data, enabling the application to be used anywhere at anytime. Because the application is running locally on the device, it can be designed to take advantage of all the capabilities of the device, ensuring a rich user experience.

Disadvantages

The design and development of an occasionally connected smart client application may be more complex than an online application. In addition, you need to consider how to deploy the application, as well as how to maintain it.

BEST PRACTICE STEPS

A successful smart client application must leverage the appropriate technology, address design challenges unique to mobile database systems, use appropriate development tools, and have a solution for remote management. Data is your most important corporate asset. Any design considerations for corporate applications should start with the data: your data must be accurate, up to date, secure, and always accessible.

Step 1 - Choose your database and data synchronization infrastructure

When creating an occasionally connected smart client application, you need to ensure that you are using database and synchronization technologies that provide the features you need. They must be designed to withstand the rigors of distributed computing at the front lines of your business. Additionally they need to provide the robustness and reliability you require for your corporate data.

SQL Anywhere has been used by thousands of companies for mobile computing projects, and it provides all the features and capabilities required to handle the most demanding applications. SQL Anywhere includes the following technologies, designed for mobile applications: MobiLink for data synchronization, and SQL Anywhere Server or UltraLite for your remote database.

MobiLink provides flexible and robust bi-directional synchronization with enterprise data, ensuring any changes made to a mobile database are reflected in the enterprise data source and vice versa. It provides built-in encryption of the database and 128-bit encryption of the communication stream. It has support for data subsetting and partitioning, and provides conflict detection and resolution, as well as error handling. Furthermore, MobiLink is scalable to thousands of remote users: it is multi-threaded, uses connection pooling, and has support for load balancing of multiple servers.

SQL Anywhere Server is a robust, full-featured SQL database, built from inception to address the unique characteristics of frontline environments, such as high-performance server and desktop applications running at remote customer sites, workgroup applications deployed in enterprise departments and remote offices, and mobile applications used by sales people or service technicians. It offers broad development tool support, and supports a very rich set of functionality, including strong encryption.

UltraLite provides the benefits of a relational, transactional-based database on resource-constrained devices. It maintains an extremely small footprint. UltraLite provides robust and reliable data storage, including transactional processing (commit/rollback) and referential integrity. It has excellent performance and provides for data indexing. It has support for Palm, PocketPC, and Symbian handheld platforms. Its tools support includes C, C++, Java, .NET languages, and UL POD for M-Business Anywhere. Strong encryption is also supported.

Other Components SQL Anywhere provides additional features, such as an application-to-application messaging solution that delivers secure and assured message delivery to mobile users.

Why SQL Anywhere: Technology considerations

The database and synchronization solutions you choose must meet your requirements. Consider the following factors related to the technology:

Wide platform support - On the server side, MobiLink can synchronize with virtually any data source, including popular enterprise class relational databases such as Oracle, SQL Server, Sybase ASE, DB2 and, of course, SQL Anywhere Server. It can also synchronize with non-relational databases, web services, web servers, applications, spreadsheets or text files. The MobiLink server can run on a wide variety of 32- and 64-bit platforms such as Windows, Linux, Unix, and Macintosh. SQL Anywhere Server remote databases run on Windows, Windows Mobile, Linux, Unix and Macintosh, while UltraLite databases can run on Windows, Windows Mobile, Palm and Symbian.

Scalability - MobiLink can be used in small setups with just a few remote databases, or it can work efficiently with hundreds of thousands of remote databases.

Small footprint - When installed with synchronization and other software, a SQL Anywhere Server database is about 6 MB in size. UltraLite has an even smaller footprint that can be as small as 400 KB.

Ease of use - The database must run without the user having to know anything about administering it, or even without the user knowing it exists. Both SQL Anywhere Server and UltraLite are designed so that the user does not need to do any administration or maintenance. SQL Anywhere includes features for self-administering, self-tuning, and remote support, making it ideal for widely deployed, zero-administration environments.

Security - The database must be completely reliable and secure. MobiLink supports a wide variety of options for securing your connections, including elliptic-curve (ECC) encryption, RSA encryption, and FIPS-approved RSA encryption. SQL Anywhere Server and UltraLite databases also support security for data, including simple obfuscation, AES encryption, and FIPS-approved AES encryption.

STEP 2 - Design your database and synchronization infrastructure

The design of smart client applications can pose new challenges for application developers. The design of both the database schema and synchronization solution must be considered up front. The database schema on the mobile device must be designed to be efficient, enabling good performance of application queries, as well as ensuring data integrity. The synchronization infrastructure must be designed to be efficient, and should only transfer data that is needed by the user. The synchronization design may need to handle upload conflicts, uniqueness requirements, and the downloading of deletions. Most organizations will start the design stage with one of the following starting points:

- The organization has an existing consolidated database and schema that is being mobilized.
- A process is being automated for the first time, so both the consolidated and mobile database schema must be designed.

Since many organizations are starting with an existing consolidated database, the particular needs for synchronization should be considered first. Developers and DBAs should consider what changes are necessary to the enterprise database to support effective mobile synchronization. These changes may include, but are not limited to:

- Addition of timestamp columns to indicate the modification date of data rows
- Addition of shadow tables to record transactions
- Appropriate column defaults

Synchronization Design Considerations

- **Synchronize only what you need.**

In most cases, data should be synchronized only if it has changed, and synchronization should only download rows that are relevant to the user. In addition, each remote database should only contain a subset of the tables and columns in the central database, and only some of those tables and columns should be synchronized. While some data should be fully synchronized, other data should be upload-only or download-only.

You may want to implement a system of high priority synchronization: time-sensitive data is scheduled to be updated frequently, but less time-critical data is scheduled to be updated at night or when the device is in a cradle. In addition, your users may benefit from a push-synchronization system where data is pushed down to mobile devices as needed.

The appropriate schema elements may need to be added to the consolidated database to support the chosen approaches.

- **Handle upload conflicts.**

Conflicts can occur when multiple mobile users update the same piece of data. If this could happen, you should define business logic to handle upload conflicts. For example, in an inventory situation you generally want all uploads to be additive. In other cases, you may want one user's input to overwrite another user's.

- **Use unique primary keys.**

The solution to upload conflicts relies on every remote database having unique primary keys. This means that each row must have a primary key that is unique not only within the database, but within the entire distributed database system.

There are a variety of solutions, including using globally unique identifiers (GUIDs) for primary keys, creating primary key pools, using composite primary keys, and using an autoincrement function to create primary keys. Which is best depends on your application. For example, composite primary keys are a simple approach but can be inefficient to search if they are large. Global autoincrement is very efficient and scales well, but cannot support self-checking identifiers.

- **Handle deletes.**

When someone deletes a row from the central database it cannot be simply deleted, because the change needs to be synchronized to all the remote databases. There are various ways to handle this, including adding a column to each table that flags rows as "deleted" or creating a shadow table of each table that holds the primary keys of deleted rows.

- **Process transactions.**

In a synchronization application, only database transactions that are committed should be synchronized. In addition, all committed transactions involving data that is to be synchronized should be synchronized, or an error should be generated.

Database Design Considerations

These considerations are important when designing the remote or mobile database schema because it may differ from the consolidated schema.

- **Define your tables.**

Use entity-relationship (ER) modeling to define the architecture of your database. Following best practices for relational databases, normalize your tables.

- **Define appropriate primary keys.**

A primary key is a column or group of columns whose values uniquely identify every row in the table. Your application must have complete control over the assignment and use of primary keys. It is usually a very bad idea to update a primary key, particularly in a synchronization environment.

- Define appropriate foreign key relationships.
Foreign keys are used to relate values in a child table to those in a parent table. A table can have multiple foreign keys that refer to multiple parent tables linking various types of information.
- Define appropriate indexes.
Indexes allow quick lookup of information. SQL Anywhere automatically creates indexes for primary keys, foreign keys and unique columns, but you need to decide whether there are other indexes that you should create manually. Once created, indexes are transparent. Indexes are important tools for efficient use of a database, but you should only define ones that are useful. Don't overdo it.

STEP 3 - Choose your application development approach

Once a plan has been established for placing data on the device and synchronizing it, you need to decide how the actual application will be developed. While you can successfully build data-driven applications using many different languages and tools, before choosing you will want to consider what platforms the application will need to be deployed to, what extra mobile-specific functionality will need to be supported, and the strengths and experiences of the development team.

Web applications

If your development experience and skills are in building web applications, you could use a tool such as M-Business Anywhere. M-Business Anywhere combines the benefits of a smart client application with the benefits of an online approach. The web application runs locally, yet changes to the web application on the server are automatically deployed to each client device. The M-Business Anywhere Application Edition includes on-device database access, allowing for full featured enterprise applications.

M-Business Anywhere provides rapid integration with existing applications and infrastructure. It offers TCO savings with web standards and tools. You can "write once, run anywhere." M-Business Anywhere provides usability and flexibility with web design. It has a smart client, always-available architecture.

M-Business Anywhere web development provides faster and more cost-effective development using established web tools and standards, a personalized UI and easy application upgrades, cross-platform support (Palm and Pocket PC), integration with existing applications, and server-based user and group management.

Mobile-specific development tools

Using a 4GL IDE that is specifically intended for building mobile applications can be the most efficient way to develop exactly what you need. Consider a Rapid Application Development tool such as PocketBuilder. PocketBuilder includes DataWindow technology to allow you to rapidly create sophisticated views of your data. It is also tightly integrated with SQL Anywhere products.

PocketBuilder allows UI development with minimal coding. It provides Painters that simplify development of database, DataWindow and synchronization applications. You can export projects from PowerBuilder into PocketBuilder so that Windows Mobile applications have a similar look and feel but a different IDE. PocketBuilder has a rich set of components (command button, picture button, checkbox, radio button, static text, list view, tree view, list box, dropdown list box, edit mask, single line edit, multi-line edit, progress bar, scroll bars, line, oval, rectangle, tabs, and so on).

A PocketBuilder application provides users with a single component for displaying and manipulating data. It has built-in SQL for selects, updates, inserts and deletes. Users can use a freeform format or a graph, grid, group or tabular format.

Language of choice

SQL Anywhere Server supports applications written in .NET, Java, ODBC, ADO, OLE DB, JDBC, PHP, Open Client, Perl, embedded SQL, and more. UltraLite supports C, C++, Java, .NET languages, and UL POD for M-Business Anywhere. In addition, SQL Anywhere and UltraLite integrate with all the major IDEs such as Visual Studio and Eclipse.

STEP 4 - CHOOSE YOUR REMOTE MANAGEMENT APPROACH

You have data on the device and it is synchronized. The application is built. Now what about deploying and managing the application on mobile devices?

Full-featured mobile device management and security

Afaria combines simple application deployment with full-featured mobile device management. Afaria manages and secures the front line of business. It goes beyond system management, with capabilities including security, process automation, and data and content management. It is optimized for dealing with varying bandwidth or intermittent connections. It is an enterprise-class, scalable solution that allows you to manage frontline devices from a single console. It supports all the devices used on the front lines of business, including laptops, handhelds, tables, smartphones, POS devices and desktops. For more information about Afaria, see <http://www.ianywhere.com/afaria>.

Web application deployment

For web application deployment you can use M-Business Anywhere. M-Business Anywhere is the industry's most scalable platform for delivering web-based content and applications to mobile devices rapidly and cost-effectively, with minimal recoding. Companies have used M-Business Anywhere to deploy a multitude of mobile applications including portals, customer relationship management (CRM), field service, logistics, data collection, inspection, help desk, and business intelligence. M-Business Anywhere can be combined with Afaria and SQL Anywhere to provide a comprehensive deployment, security, and management platform.

For more information about M-Business Anywhere, see:
<http://www.ianywhere.com/whitepapers/index.html#mbus>.

CONCLUSION

Data is a significant corporate asset. You must protect it by keeping it reliable and secure in the field. Designing and managing your data is the first step in mobilizing your systems.

Too often, first-time developers of mobile applications do their application development first, leaving their database and synchronization design to the end. This reversal of steps introduces the risk that any changes to the application required by the database or synchronization design will delay the application roll-out.

For many business applications, synchronization is the key to making sure your data is brought reliably and securely back to the enterprise. Synchronization is also the key to having mobile applications operate as an extension of your enterprise data systems. Mobilink synchronization provides a way to create a rich user experience:

- The user application is always available.
- Users can access all sorts of enterprise data.
- The technology “just works” and is transparent to the end user.
- The data is secure and cannot be compromised.

By choosing the best technology and application design for your data needs, you can ensure that your mobile applications give you a competitive advantage and increase the overall competitiveness of your business.

IANYWHERE SOLUTIONS, INC.
WORLDWIDE HEADQUARTERS
ONE SYBASE DRIVE
DUBLIN, CA 94568-7902
U.S.A.

CONTACT_US@IANYWHERE.COM
NORTH AMERICA
T 1-800-801-2069
1-519-883-6898
EUROPE, MIDDLE EAST, AFRICA
+44 1628 597 100
ASIA PACIFIC
+852 2506 8700
JAPAN
+81 3 5210 6380