Developing Mobile Applications Using SCO UNIX and SCO Mobile Server

Jonathan Schilling  
jls@sco.com
Agenda

SCO Mobile Server
  structure
  advantages
SCO Mobile Software Development Kit
  purpose
  components
Examples of SCO mobility applications
  FCmobilelife
  HipCheck
SCO Mobile Server – What is it?

- SCO Mobile Server is a *Mobile Application Platform* that runs on SCO UNIX
- SCO Mobile Server significantly reduces time and cost of
  - developing mobile solutions
  - extending existing business applications to mobile devices
- Smartphones, PDAs, feature phones, SMS-only
- Web browsers, iPhone browser, native PC desktops
- **Server side** software components required for high performance
- Administrative Web interface (Portal)
- **Client part of services** (Applications on mobile devices)
- A SCO Mobile Server API set
- **Backend integration part of services** (Connectors, Agents)
- SCO OpenServer 6 or UnixWare 7 provide the base
SCO Mobile Architecture

Client side
SCO Mobile
Server
Backend integration
SCO’s Vision for the Mobile Marketplace

- A mobile application platform for
  - Mobilizing existing applications *OR*
  - Creating new mobile applications

The SCO Mobile Server provides the Essential Infrastructure for Linking Mobile Devices to Backend Systems
Secure, reliable connections to existing applications of all types
Client application versioning
Administration of custom applications
User, group & account model
User authentication
Secure https transaction processing
Internationalization/localization support
Load balancing, scalability, reliability
Back-end system connector/agent framework
Support for both traditional and subscription-based solutions
Google CEO Schmidt quote:
- My prediction would be that Web 3.0 will ultimately be seen as applications which are pieced together. There are a number of characteristics: the applications are relatively small, the data is in the cloud, the applications can run on any device, PC or mobile phone, the applications are very fast and they're very customizable. Furthermore, the applications are distributed virally: literally by social networks, by email.

Another description floating around:
- Web 1.0: read-only
- Web 2.0: read-write
- Web 3.0: read-write-execute

SCO Mobile Server
- has some characteristics in these directions …
What do you need for a mobile business application?

A central service platform like SCO Mobile that provides:

licensing (how many users can access a mobile application?)

User authentication - which users (phones) are permitted to access different mobile applications

Application specific configuration (e.g. Who is allowed to do what?)

The SCO Mobile server can run at your customer site, an IT Service Partner, or at an Application Service Provider (e.g. SCO)
What do you need for a mobile business application?

**Developer or partner companies like SCO with Know How to**

- develop the mobile client on the phone
- develop the backend integration, …

(with tools like Visual Studio, Eclipse etc.)

The **SCO Mobile DevKit**

It provides APIs in order to access SCO Mobile Server services, documentation, phone emulators, …
The basic licensing model is based on a **Initial Perpetual License** followed by **Annual Software Maintenance fees** after the first year.

Pricing is based on the number of users who can authenticate to the system.

Adaptable to **SCO-hosted**, **Reseller-hosted**, or **customer-deployed** scenarios.
Types of SCO Mobile Digital Services

- Self-Contained Service
- Business Integration Service
- Monitoring Service
- Mobile Database Service
- Two-way SMS Service
Self-Contained Service

- A new application
  - Does not connect to existing application
  - No agents necessary
- Subscriber-centric
  - Also groups, contacts, etc.
- Most logic on SCO Mobile Server
- Stores application data in server database
  - Same machine or co-located dedicated db machine
- Can be tied to “landing pages” or “stores”
- Examples: FCmobilelife, Shout Postcard, WIGS
Business Integration Service

- Connects to existing application
  - Therefore, has connectors or agents
- Purpose is data collection, data reporting, transaction processing
- SCO Mobile Server becomes mostly a “switch”
  - Its local database is mostly just used for routing
- Application is client- and agent-centric
- Has web admin pages to manage agents
- Examples: Musco Foods, Carlamobile, DTP
- Specialized variant of business integration service
- Tracks vital business and system metrics
  - View health indicators of business or system
  - Receive alerts based on user-set triggers
  - Take actions to correct problems
- Combines SOA with EDA
- Examples: HipCheck (for Operating Systems)
  - HipCheck Source Code Product available for others to customize it to particular environments
- Keep full application database on mobile device
- Periodically synch with back-end server
- All sorts of data capture are possible
- Biometric input devices with generic interfaces
  - e.g. fingerprint scan
- Examples: Mobile local census info
Two-way SMS Service

- Existing application sends SMS to phone, phone user replies by SMS, gets routed to application
- No client program on phone needed
- Any phone can be used
- Appropriate for well-defined, limited user interactions
- Must supply SCO Mobile Server plug-in for two-way SMS provider chosen
- Examples: SCO Mobile “TeamLink”
  - For a coach coordinating responses from team members
  - For confirming doctor/dental appointments
Best platform to run on
Available as package SCOms
OpenServer 6.0.0 mp3 or later, UnixWare 7.1.4 mp3 or later
pkgadd format for both
Available for download from
http://www.sco.com/support/download.html
Release notes are online
Software development kit for SCO Mobile Server
- Provides libraries of reusable building blocks
- Provides plug-ins, templates for popular IDEs
  - Visual Studio, Eclipse, NetBeans …
- Free for development use
- Available in open beta
- Installs on Windows desktops/laptops
- Contains 5-user-licensed SCO Mobile Server
- Contains API libraries for client and server development
- Contains several sample applications
- Contains extensive documentation
■ Development environments not included
  ■ Licensing restrictions, size issues, version control
■ Download and install them yourself
  ■ Eclipse – used for Java service development
  ■ Ant – used for Java service builds
  ■ Visual Studio Mobile Edition – used for Windows Mobile clients
  ■ Sun Wireless Toolkit – used for Java ME clients (simple)
  ■ NetBeans – used for Java ME clients (full-featured)
- Java Micro Edition
  - Many variants (even in MIDP 2.0)
  - Many phones, many phone OSes
  - Portability can be difficult to achieve
  - Subset functionality for cross-phone support
  - SCO Mobile SDK provides library for account mgmt, Mobility Server communication, etc.
• Sun Wireless Toolkit is basic environment
• NetBeans for Mobility is full-featured and better
  • Open source for Sun
• BlackBerry has own Java Development Environment (JDE)
  • Add JDE as custom emulator to NetBeans
  • Configure both BlackBerry and “normal” Java environments to NetBeans
  • Now can switch back & forth with same source base
- Views for authentication and management of multiple accounts
- Lightweight, extensible, highly portable subsystem for creating mobile user interfaces
- Portable routines for network access and sending commands to SCO Mobile Server
Windows Mobile .NET/C#
  - Four OS subvariants
  - Treo 700, Samsung 730i, ...
  - Visual Studio for Mobile - best dev env and functionality
    - SCO Mobile SDK template
  - SCO Mobile SDK provides library for account mgmt, Mobility Server communication, etc.
  - Client apps will also run on PC Windows with some rearrangement and configuration
- Classes to manipulate and authenticate accounts
- Classes for custom controls and building custom mobile user interfaces
- Classes to access and manipulate SQLite database tables
- Utility classes for datatype parsing, error handling, etc.
- Classes for sending and receiving HTTPS requests to SCO Mobile Server
- Lightweight SQL ADO.NET database for persistent data
iPhone client support libraries not in SCO Mobile SDK yet ...

But iPhone support has been validated by ongoing FCmobilelife work

Here are some of the key aspects ...
- Objective C programming language
- Cocoa framework
  - From Apple, previously “NeXTSTEP”
  - Cocoa Touch Layer
  - UIKit
  - Media Layer
  - Audio Toolbox
  - Audio Unit
  - Core Audio
- C
- Core Foundation (Dates, Calendars, XML Parsers... Used for communication to server)
- System Configuration
- Core Services Layer (Also used for server communications)
Client Development Environment – web clients

- Client runs in a web browser
- Many technology choices
  - Flash, using Flex toolkit
  - Javascript, using a variety of approaches
  - PHP
  - Mixtures
    - FCmobilelife is part PHP, part Flash
- Communication with SCO Mobile Server is same
Client Development Environments: Other

- **HB++**
  - Treo 650 and PalmOS
  - Best-looking UI
  - Programming can be difficult

- **SuperWaba**
  - Java variant for high-end phones
  - Good-looking UI, good portability
  - Little known, support questionable

- **Symbian/BREW**
- **AJAX**
  - Conventional web browsers
  - cf. WebFace, Google

- **Others**
  - Other C++ variants
  - Python
  - Anything that can do http name/value pairs can be used, but no direct SCO Mobile SDK support
- Java based
- OS neutral
  - Develop code on Windows, deploy code on SCO UNIX
- Service processor
  - Uses Java EE app server infrastructure (JBoss)
  - Almost completely transparent to application code
  - Don’t have to be a general Java EE expert!
- Web admin pages
  - Mobility Server uses Java Tapestry framework
  - Applications often use PHP (or whatever is familiar to you)
- Mobility server database access
  - Data object layer, database common to both service and admin
  - EJB3 Persistence, Spring/Ibatis, native Ibatis, JDBC …
- Hello, world
- Meter Reading
  - Showcases simple Java clients, role of agents
- Car Dealer Service
  - Showcases web admin pages
- Echo SMS Service
  - Showcases two-way SMS
  - Restricts user input, but …
  - Gives broadest range of client support – just about any phone
- Installation and Release Notes
- Getting Started Guide
  - Includes instructions for building and deploying sample apps
- Developer’s Guide
  - Overview, sections on client and service programming
- API doc
  - Service APIs in javadoc format with commentary
  - Client programming APIs
Connectors run on SCO Mobile Server

- Communicate directly with existing back-end system
- Use for pulling in subscriber/group info from HR DB
- Some applications or database products provide direct web service access
- Use direct WS, JCA, JDBC, ftp … may avoid agents
- Firewall issues
- Solution will be unique to each customer/partner/application/technology
Agents run on an existing back-end system
- Agent access is by SOAP/XML-based web services
  - Supported by many OSes, languages, tools
  - SCO Unix provides in Java, C, C++, Perl, PHP
- Expose apps as services
  - Key to mobilizing existing applications
- Write database code as service, construct intermediary flat file or mini-db …
- Firewall issues
- Solution will be unique to each customer/partner/application/technology
Why not … use phone browser as the client?

- Characteristics of phone browsers
  - Tend to poor presentation layout
  - Can be unreliable
    - Treo Blazer, for example
  - Can be slow
    - Full page refreshes, no state
  - Difficulty in accessing phone features
    - Camera, fingerprint scanner, voice recorder, etc.
  - Difficulty in maintaining local data
    - A ‘real’ client supports local data, encryption, store-and-forward, etc.
Why not …
go directly from client to back-end server?

- This can be done as a one-off …
- But loses many services that SCO Mobile Server provides
  - Common user, group & account model
  - Common authentication
  - Common https transaction processing
  - Common data caching, store-and-forward
  - Common load balancing, scalability, reliability
  - Common connectors and agents framework
  - (Optional) subscription-based billing model
  - Ability to have apps invoke other apps as services
- Powerful mobile client, web client for real-time collaboration for small groups
- Includes calendaring, scheduling, task management, and multimedia
- For small businesses, families, teams and clubs
- Delivered in partnership with FranklinCovey
FCmobilelife client (Windows Mobile)
FCmobilelife client (Blackberry)
HipCheck is an SCO Mobile application
Allows you to monitor the health of your UNIX and Windows systems from your mobile devices
- View current state
  - processes, users, disks, CPU, memory, printers, services, etc.
- Set triggers to get alerts about critical system conditions
  - Service stops running, disk gets low on space, etc.
- Take corrective actions
  - Kill runaway process, restart crashed service, reset password, etc.

Hosting and business model choices
- HipCheck service may be hosted by SCO – subscription model
- Or deployed by reseller or end customer – conventional model
- The bridge between SCO’s UNIX and mobile product worlds
HipCheck Clients

The HipCheck Client – Phone

Windows Mobile Client

Java based, e.g. PALM Treo

BlackBerry
Web interface for Admin Tasks:

- **HipCheck subscribers and groups**
- Configure HipCheck user, account, groups, …

**HipCheck specific administration**

- Licensing
- Create/modify monitored system definition
- Set/modify privileges of subscribers for a system
- Status check on monitored systems
HipCheck agents

The HipCheck Backend integration

- Agents run on the monitored server
  - SCO OpenServer 5
  - SCO OpenServer 6
  - SCO UnixWare 7
  - SUN Solaris 8+
  - Linux
  - Windows 2000 Server
  - Windows 2003 Server
  - Windows Vista
  - Windows XP Pro
HipCheck Security – example of mobility security

- **Client to SCO Mobile Server**
  - Me Inc subscriber must login and be authenticated
  - Me Inc account must be eligible for HipCheck
  - Communication is over https

- **On SCO Mobile Server**
  - Administrator must grant access to specific monitored systems
  - Administrator must grant additional access to set alerts or take actions or add other subscribers
  - Administrator must supply root/admin account, password of monitored system

- **SCO Mobile Server to Agent**
  - Communication is over https with certificates
  - Monitored system is typically protected by firewall
HipCheck availability

- Stable, mature product
- HipCheck 1.1 released early 2008
- Included many new features
  - Audit trail
  - Improved Windows mobile/Windows native UI
  - Support for Windows Mobile 6
  - Support for Linux agents
  - Execute any command from phone and see output
  - View software inventory
  - Active Directory authentication support
  - Japanese client localization
- SCO Mobile Server supports building several different types of mobile digital services
- Same tools, libraries, methods, and deployment platform are used for all of them
- Superior to other approaches for mobilizing applications
- FCMobileLife and HipCheck are prime examples of the capabilities of the SCO Mobile Server
Q & A