

## Comparing Mobile Applications Platforms

### Introduction

The Net Caboodle approach to developing software services for mobile and lightweight devices is fundamentally different to the common crop of mobile development tools and systems. In this data sheet we compare the Net Caboodle approach with some current approaches that use Web Service, or other xml based protocols.

The current Web Service style systems for connecting mobile devices (for example the JSR172 standard) are targeted at writing mobile clients that connect to services running on the Internet. For example, Amazon.com provide a web services interface to run queries, and return replies formatted in an html/xml style. However, there are many types of mobile software application for which this approach is inappropriate, and as a result the Net Caboodle approach is squarely targeted at connecting to enterprise-level services with a much faster, lighter and more secure protocol, along with tools that automatically write the software to connect the mobile application directly to services running in the corporate network.

### Net Caboodle Background

The Net Caboodle system is an enterprise class toolset that provides the ability to connect software services (SOA) that are running in the corporate network directly to mobile devices, using both Java and the Microsoft Windows Mobile platform, without having to write any 'connection' or 'glue' code between the servers and device. (See the Java & Windows box)

The Net Caboodle toolset is the result of a number of years of research and development and has been in production use in a number of mobile applications for over two years. Net Caboodle servers run round the clock providing the backbone of revenue producing mobile services such as Email, Webcam access, Push to Talk and SMS gateways.

Device servers on the open internet via gateways into all service providers that allow open internet access, including Vodafone, Orange, T-mobile, Cingular and other local network operators. The system has been tested and deployed and is generating subscriber revenues, running transparently over packet and pseudo packet protocols including 2.5 and 3G networks in over 70 countries.

The Net Caboodle handset users include over 150 of the most popular current handsets including to Nokia, Motorola, Sony Ericsson, Siemens and HTC. In the following sections we look at various features of the Net Caboodle platform and contrast these with Web Services (e.g. JSR172) approach.

#### Java Mobile & Windows Mobile

Two of the most popular environments on which to run mobile software. The Net Caboodle platform can automatically produce code to run on both types of devices and connect transparently to services running within your corporate network.

## Compatibility

Despite claims that the Web Services approach is standard for all platforms, it is necessary for the device to support and run the JSR172 API, [or the appropriate XML parsing libraries on Windows Mobile] therefore limiting the number of potential devices the mobile application can run on. Net Caboodle has targeted the mass Java (MIDP-1.0 and above) and Windows Mobile (version 5 and above) phone market by using only the libraries 'built into' the platform.

Because the Web Services approach requires that all messages take the form of Web Services (XML) messages this is the only interface type that can be used to communicate with the device. With Net Caboodle it is also possible to use Java Remote Invocation Services (RMI), Jini Services, CORBA for cross language connectivity, and plain old Java objects (POJOs)

## A Lightweight Protocol

The Net Caboodle platform uses a lightweight binary protocol to communicate between the device and network services. This has a number of advantages, not least of which is that that amount of network traffic to specific messages is greatly reduced.

In the table we show an examples of messages that have been encoded in both the Net Caboodle and the standard JSR formats and compare the size of the messages.

Hello World Service	Web Service Bytes	Net Caboodle Bytes
Request size	313	14
Response size	233	28

So not only is the data smaller but for any given network bandwidth the message will be faster to and from the device by the same proportion. This is very important in trading (Betting, Spread Betting, Equities, Options, etc) applications where time is of the essence. It also reduces the chance of a message being 'split' by a network outage or disconnect by the 'train in the tunnel' problem.

Because the protocol is much closer to the encoding that the software on the device uses to store the messages, there is a much lower overhead in transforming the data, both on the servers and on the device. This yields much faster server to mobile screen performance than heavier protocols.

From a deployed cost perspective, the Net Caboodle system uses the network in a far more efficient way and in a context in which network bandwidth is chargeable (3G tariffs for example) this can mean a 10 times reduction in bandwidth charges, making certain applications economically viable, when previously they may have been ruled out because either network customers would not bear the high bandwidth costs, or the network would not subsidise the network costs.

## Security - Authentication & Encryption

The Net Caboodle toolset has built in (and customisable) features for security and authentication available 'out of the box'. In contract each web service and web

service mobile client application has to have this functionality 'retro-fitted' for each application that is developed.

The Net Caboodle, platform has a plug-in model for ensuring the 'identity' of mobile devices and individuals connecting to the network and a simple and fast encryption. In keeping with our overall design philosophy, we provide small and simple plugins for both authentication and encryption.

In JSR 172, encryption is provided via HTTPS which itself requires at least MIDP 2.0 on Java devices [and Equivalent on Windows Devices] which also limits the number of devices usable by clients.

## Transactions

Transactions are used by systems designers and implementers to ensure that a particular event, such as a credit card payment, is either completed (committed) or never takes place (aborted). This is vital, again for trading systems of any kind, where exchange of goods or services for payments takes place.

In comparison to the Web Services model where that is no simple built-in support for transactions, the Net Caboodle system has a pluggable transactions model suited to mobile applications, in which leased Transactions are used to ensure that even in the event of a net work disconnect a transaction will abort leaving none of the parties in a transaction 'short changed'

## Developer Tools and Development Environment

The Net Caboodle development tools are provided as an Integrated Development Environment (IDE) plugins for both Eclipse and NetBeans plus Visual Studio plugin on the road map The developer system runs a copy of the Net Caboodle gateway server within the IDE so that new Mobile Applications can be developed and deployed from within the IDE, as part of the standard development cycle.

The Net Caboodle tools automatically generate all of the code required to connect enterprise services out to the mobile device and this code is comparable is size to the code generated to support web services on the device, bearing in mind that the Net Caboodle code is not dependent on any 'extra' on device libraries or code above the base Java or Windows Mobile platform.

The Net Caboodle generated code is a JAR or DLL for Windows Mobile and is linked into an IDE project, meaning that the mobile client code is clear, clean and precise. In contrast some IDE generated code for Web Services based applications can be complex to understand because the flow of both a mobile client and the server-side components is not natural or simple, which may be fine for a trivial prototype, but can generated maintenance headaches as the complexity of the code base increases. We understand that these are somewhat subjective measure and so we encourage developers to download the free trial version of the developer tools and make their own judgements.

### Mobile Trade and Commerce

The use of both Secure and Transactional services are vital in order to make mobile trading systems work. You need to be able to identify the person or business entity your are dealing with and have a very clear transactional model to develop trust in the underlying payment or money transfer system.

## Deployments

To deploy Web Service style mobile systems it is necessary to deploy them on an Enterprise Java (JEE) Application Server for, at least, servlets and JSPs, with the implied support and configuration file requirements. In contrast the Net Caboodle tools automatically produce a deployable service binary (a pre-configured instance of the Net Caboodle Gateway) which can be used for any number of deployed servers, for example, in a failover scenario.

The Net Caboodle mobile gateway does not embed IP addresses (server locations) within the deployment configurations which makes it much more flexible than the JEE alternative. This means that the location of servers can be changed within the running system, because the mobile devices are not statically bound to a server location.

Both systems can use the Apache HTTP server for front ending HTTP traffic and as a result share the scalability and flexibility this brings. However use of heavy weight protocols on both the Device network side and on the corporate network side, has been seen to rapidly inhibit scalability of the system as a whole.

## Cost

We make the cost of a single server deployment of the Net Caboodle Gateway comparable with the equivalent cost of a Web Services systems deployed on any of the Commercial JEE Application service.

However for medium to large scale deployments we have found that because the Net Caboodle gateway runs faster and lighter than the JEE equivalent, fewer servers are required to support an equivalent number of concurrently connected clients. This not only means reduced server licensing costs but also reduce deployment complexity and infrastructure (server hardware) requirements

Another major on going cost reduction derives from the reduction in complexity of the software code base required to support the Net Caboodle platform hence reduction of maintenance and complexity of the resulting system. This means that developers can add features and fix bugs in your business code rather than maintain and extend protocol and glue code.



Net Caboodle Limited (UK)  
59-60 Thames Street  
Windsor  
SL4 1TX  
United Kingdom

Tel: +44 (0) 1753 272 312  
Fax: +44 (0) 1753 272 313

Copyright notice

The information within this document remains the sole property of Net Caboodle Limited. No part of this document may be copied or reproduced in any form or by any means, and the information contained within it is not to be communicated to a third party, without the prior written consent of Net Caboodle Limited. All trademarks remain the property of their respective owners.