Pronto: Mobile Gateway with Publish-Subscribe Paradigm over Wireless Network

Eiko Yoneki

University of Cambridge Computer Laboratory,
William Gates Building, J J Thomson Avenue
Cambridge CB3 0FD, UK
{Eiko.Yoneki}@cl.cam.ac.uk

Abstract. Pronto, a messaging middleware system for mobile applications provides both centralized and decentralized models. It addresses design issues in mobile computing, including data optimization and disconnected operation. Pronto consists of a lightweight message-oriented middleware (MOM) client and an intelligent gateway serving as a message hub with store-and-forward messaging.

Computing devices have become increasingly mobile at the client end, and the diversity of clients and networks creates complex environments for mobile applications. In a mobile/wireless environment, devices have a small ROM/RAM footprint, high latency, and low bandwidth. Connections are frequently interrupted. Middleware communications service is especially important for integrating such hybrid environments into coherent distributed systems. We developed Pronto, a messaging middleware system for mobile applications over mobile/wireless networks, to address these.

Evolution of message-oriented middleware

Intercommunication has been commonly achieved using directed links between tightly coupled senders and receivers such as in RPC or CORBA; the message destination must be known at the time of sending, which is difficult with changing destinations or varying numbers of recipients. By contrast, Message Oriented Middleware (MOM) encourages loose coupling between message producers and consumers with a high degree of anonymity, thus removing static dependencies in distributed environments. MOM's characteristics (intuitive programming model, latency hiding, guaranteed delivery, store-and-forward) are highly appealing for mobile applications.

MOM is acknowledged in the business domain through the provision of Sun's Java Message Service (JMS) API [10] as a common interface. JMS offers the publish-subscribe paradigm and expands previous messaging capabilities. Existing MOM software has been rapidly integrated under the JMS API such as IBM's MQSeries [3], Softwired's iBus [11], and BEA's WebLogic [12].

A complementary distributed system, based on a peer-to-peer network, emerged independently of the enterprise domain. The peer-to-peer interaction model facilitates sophisticated resource sharing between peers over the edges of the Internet. There have been research efforts to build messaging systems over peer-to-peer networks. Scribe [9] from Microsoft is a topic-centric messaging system based on Pastry [8]. The Siena project [1] created a distributed event service, while Hermes [6] offers an event broker system. Gyphon [2] is a distributed topic and content-based message brokering system. Narada [5] is based on the Grid Event Service implementing JMS.

Integrating of messaging systems and mobile computing

In the enterprise domain, messaging is increasingly popular for communication, because it supports decoupled interactions. However, intranets with centralized administration are still predominant. Recent extensions to mobile computing add an edge server to manage mobile devices. This approach is not fully scalable to dynamically extended businesses over Internet services or mobile computing.

Because of the power of the publish-subscribe paradigm, peer-to-peer network-based messaging will be most promising to establish communication platforms over wide area networks. However, support for mobile computing is not yet sufficient. Most research is focusing on integrating ad-hoc networks

ACM/IFIP/USENIX International Middleware Conference (Work in Progress), June 2003.

into peer-to-peer networks, but this will not support mobile computing in its full generality. Decentralized messaging systems are still immature and fragmented.

Most JMS products implement centralized server models. To provide rich JMS functionality, especially persistent message delivery, servers require databases for message storage. Thus far, none of the commercial products have successfully implemented JMS in a decentralized model with enterprise level messaging functionality.

Integration of messaging systems in both centralized and decentralized models including mobile environment requires architecting messaging system from a unified viewpoint.

MOM design issues in mobile computing

Given the characteristics of mobile devices and wireless networks, more work is required for optimal performance. Important design issues are described below:

- Wireless networks become increasingly packet-oriented. With a bearer like GPRS **(**General Packet Radio Service**)** or UMTS **(**Universal Mobile Telecommunications System**)**, users typically pay only for the data they communicate. Reducing data size for transmission is crucial.
- A middleware should provide an interface to applications that allows maintaining communication during disconnected operation. Dependable caching is essential.
- A data source can be interpreted in different formats and semantics, depending on the specifications of mobile devices and wireless networks. Semantic transcoding technology [4] will allow more efficient data flow.
- There are various bearers such as 2G, 2.5G, 3G, Bluetooth, and IEEE 802.11, and many devices are non-programmable. A middleware needs to offer an interface that provides a communication abstraction.
- An ad-hoc network is a dynamically re-configurable wireless network without fixed infrastructure and without requiring intervention of a centralized access point.

What Pronto offers

Pronto [13] proposes to integrate mobile environments over MOM based on JMS [10], and it provides solutions for mobile application-specific problems such as resource constraints, network characteristics, and data optimization. A lightweight **MobileJMS Client** is designed to adapt JMS to mobile environments in both centralized and decentralized models. The specifications and interfaces of JMS are complex, but not all functions are required for mobile environments. For example, *Durable subscription* is implemented but not all message types are implemented. As an extension of JMS API, **Message Selector** is added to support content-based subscription. *Message Selector* is a filter for a topic, which provides greater flexibility for applications with coupling between producers and consumers.

Serverless JMS is a novel decentralized version of MobileJMS Client using IP multicast [7] as transport mechanism. This model will perform best in two network environments: temporal group communication over ad-hoc networks and high-speed transmission of many messages for workload distribution. IP multicast allows for efficient transmitting of messages from one publisher to many subscribers. The number of subscribers can increase without impacting network traffic. A publisher acts as a temporary server and keeps a subscription list. However, some JMS features such as Durable subscription were omitted in the current implementation, given the nature of the network model and IP multicast protocol.

We introduced an intelligent **Gateway** [14] in Pronto as a message hub using store-and-forward messaging; giving powerful optimization of data reduction and transformation. Gateway is also a MobileJMS Client to serve as a proxy of a message controller. This allows construction of a distributed messaging system over JMS servers through cascading gateways. Gateway is designed as a framework to perform plug-in functions for caching, semantic transcoding, device specific transport, and message transformation, as well as supporting disconnected operations. Semantic transcoding offers more than simple data reduction: the information is made more abstract (providing compaction), and data should be evaluated whenever necessary. Here, the data are linked to an annotation; annotation can be a document summary, a linguistic description of the content for voice synthesis, or gray-scale/low-resolution image data.

SmartCaching component is designed to provide generic caching, a central function for message storage in Gateway. It offers *Pull, Subscribe*, and *Snapshot* services. The *Subscribe* service provides asynchronous notification of cached data to client applications, and applications do not need to pull data that have been requested before. Using the *Subscribe* service client applications can be event-driven and active. *Snapshot* provides a special period for mobile applications to obtain the last cache image after disconnection.

Gateway and SmartCaching are key technologies for improved messaging among mixed mobile-tier environments in dynamic connectivity scenarios. The Gateway and SmartCaching APIs are independent from JMS API.

Deployment in a centralized model

Figure 1 gives an overview of a distributed system based on Pronto; different deployment possibilities are illustrated:

1. **Application with MobileJMS Client:** An application in a mobile device uses MobileJMS Client; it communicates directly with the JMS server.
2. **Application with Local Gateway:** An application in a mobile device uses *LocalGateway API*; *LocalGateway* runs as a separated thread or within the application and performs caching and transcoding through plugged-in components.
3. **Application with Remote Gateway:** An application in a mobile device uses *RemoteGateway API*; *RemoteGateway* runs as a separate process, and RMI-based transport is currently implemented.
4. **Non-Programmable Devices with Remote Gateway:** Non-programmable devices require *RemoteGateway* to perform proper transportation and message transformation for the target device; remote Gateway represents every subscriber and publisher for the non-programmable device.

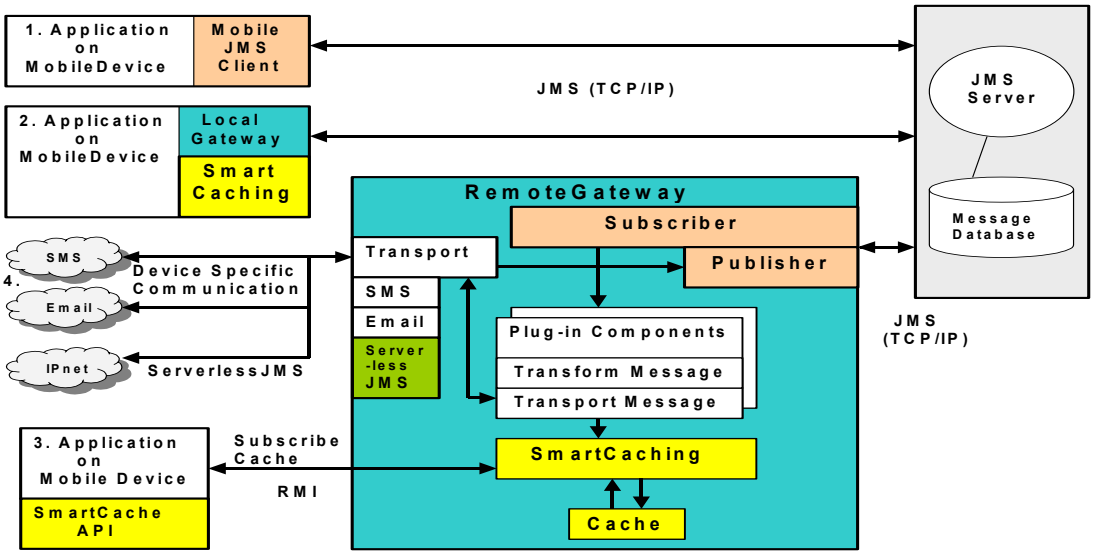


Figure 1. System Overview with Pronto in a centralized model.

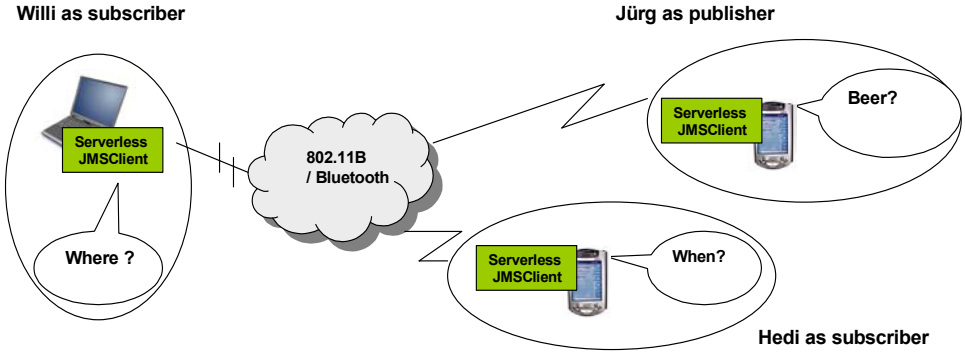
Disconnected operation

We designed the following approaches in Pronto for disconnected operation:

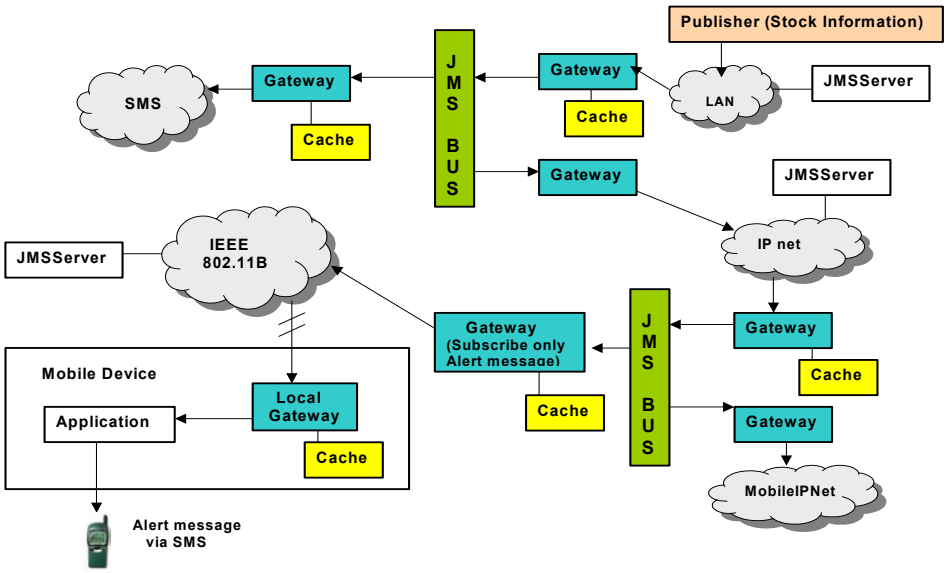
- **Durable subscription in JMS:** Non-durable subscriptions last for the lifetime of the subscriber object; a subscriber can be durable by registering a durable subscription with a unique identity.
- **Gateway Cache:** Gateway maintains the cache even if applications are inactive; applications can use *Pulling, Subscribe,* and *Snapshot* services on appropriate occasions (e.g., an application spawns a background activity that synchronizes the on-device messages when connected; the application can use cached messages locally when disconnected).

Deployment in a decentralized model

- **Chat system over Ad-hoc Network**
  Figure 2-(1) shows a chat system using Serverless JMS over a temporal network. Serverless JMS in the publisher uses the auto-discovery function to detect subscribers, and maintains subscriptions. This does not require any central access point.
- **Gateway Cascade**
  Multiple Gateways can distribute messages to the target Gateways, from where messages are sent to the devices. In Figure 2-(2), JMS bus is a Serverless JMS over a high-speed bus, which can be LAN or WAN based if the routers allow IP multicast. A combination of Gateway and JMS bus offers powerful message flow control for optimization.



(1) Chat System over Ad-hoc Network



(2) Gateway Cascade

Figure 2. Deployment Examples with Serverless JMS.

Application example: A mixed media chat system

All clients subscribe to the topic 'Chat'. A Gateway residing in the mobile laptop contains plug-in components *Voice Synthesizer* and *SMS*. iPAQ devices monitor the chat through the Gateway cache. Additional devices can join the chat anytime, without any change of the other components.

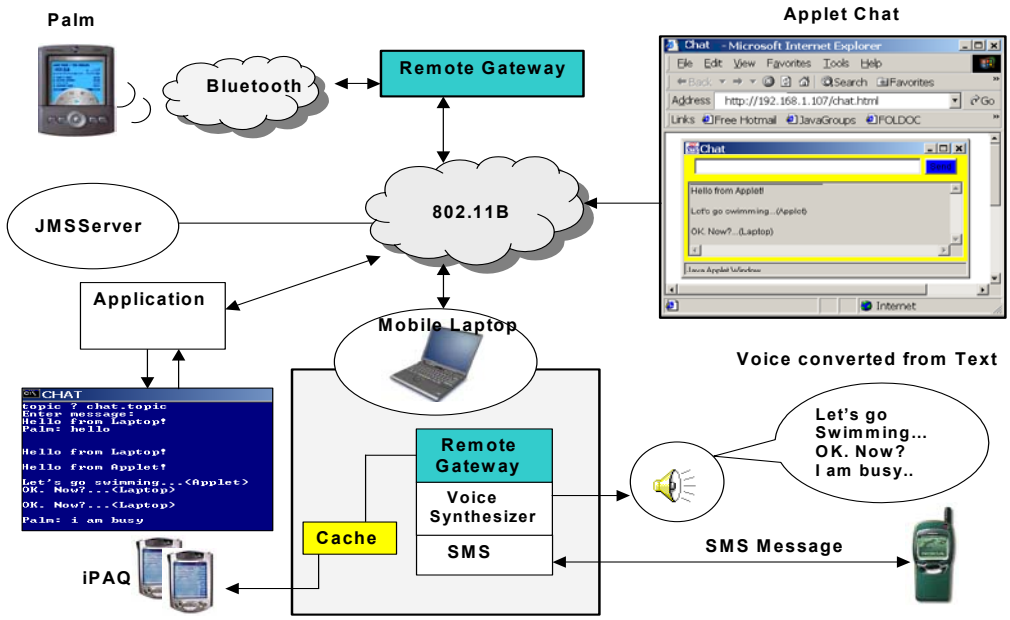


Figure 3. Application Example: Mixed Media Chat.

Future of Pronto

This article shows various design issues for a messaging system in mobile environments. Pronto aims to extend messaging over a wireless network and solve inherent problems. Pronto will be extended in directions described below:

Topics for publish-subscribe have to register publicity and are used by other applications as remotely accessed distributed objects. A standard API for publishing, managing, and accessing public references to distribute functionality over a mobile environment will be critical. This includes security aspects such as encryption, authentication, and access control on distributed objects.

We are currently working on an event-based distributed system over a peer-to-peer network using a multi-event broker model. The event broker approach is attractive for integration of generic event-based systems with mobile computing, because it allows heterogeneous messaging systems to be combined under a unified interface. In particular, constrained mobile devices can take advantage of resource sharing from middleware systems. Pronto will be integrated with the event broker system. We will focus on research to abstract messaging in hybrid mixed environments.

Acknowledgment

I would like to thank Jean Bacon and Jon Crowcroft (University of Cambridge) for constructive comments and suggestions.

References

1. A. Carzaniga et al., Content-Based Addressing and Routing: A General Model and its Application, Technical Report,
   University of Colorado, 2000. http:// www.cs.colorado.edu/~carzanig/papers/cucs-902-00.pdf.
2. Gryphon: Publish/Subscribe over public networks, IBM Research,
http://researchweb.watson.ibm.com/gryphon
3. IBM MQ Series, http://www.ibm.com/software/mqseries/.
4. K. Nagano, Semantic Transcoding: Making the World Wide Web More Understandable and Usable with External Annotations, IBM Tokyo Research Laboratory, 2000.
5. Narada Event Brokering System, http://grids.ucs.indiana.edu/ptliupages/projects/narada/.
6. P. Pietzuch and J. Bacon, Hermes: A Distributed Event-Based Middleware Architecture, *Proceedings of the 1st International Workshop on DEBS'02 with IEEE ICDCS, 2002.*
7. RFC1112, Host Extensions for IP Multicasting, http://www.rfc-ditor.org/rfc/rfc1112.txt.
8. A. Rowstron and P. Druschel, Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-
   Peer Systems, *Proceedings of the 3rd Middleware Conference,* 2001.
9. Scribe, http://research.microsoft.com/~antr/SCRIBE/
10. Sun Microsystems, Java Message Service (JMS). http://java.sun.com/products.jms/.
11. Softwired iBus Messaging, http://www.softwired-inc.com/.
12. WebLogic, http://www.bea.com/products/index.shtml.
13. E. Yoneki and Jean Bacon, Pronto: MobileGateway with Publish-Subscribe Paradigm over Wireless Network, Technical Report UCAM-CL-TR559, University of Cambridge, 2003. http://www.cl.cam.ac.uk/TechReports.
14. E. Yoneki and J. Bacon, Gateway: a Message Hub with Store-and-forward Messaging in Mobile Networks, *Proceedings of the 1st International ICDCS Workshop on Mobile Computing Middleware (MCM03), 2003.*