SyncML

Building an Industry-Wide Mobile Data Synchronization Protocol

SyncML White Paper

Version 1.0

Abstract

The popularity of mobile computing and communications devices can be traced to their ability to deliver information to users when needed. Users want ubiquitous access to information and applications from the device at hand, plus they want to access and update this information on the fly.

The ability to use applications and information on one mobile device, then to synchronize any updates with the applications and information back at the office, or on the network, is key to the utility and popularity of this pervasive, disconnected way of computing.

Unfortunately, today we cannot achieve these dual visions:

- Networked data that support synchronization with any mobile device

- Mobile devices that support synchronization with any networked data

Rather, there is a proliferation of different, proprietary data synchronization protocols for mobile devices. Each of these protocols is only available for selected transports, implemented on a selected subset of devices, and able to access a small set of networked data. The absence of a single synchronization standard poses many problems for end users, device manufacturers, application developers, and service providers.

SyncML is a new industry initiative to develop and promote a single, common data synchronization protocol that can be used industry-wide. Driving the initiative are Ericsson, IBM, Lotus, Motorola, Nokia, Palm Inc., Psion and Starfish Software. SyncML initiative is supported by several hundred leading companies in the industry, and additional companies are welcome to join the open initiative and participate.

The SyncML initiative expects to release an open protocol for data synchronization, called SyncML, mid 2000. A wide variety of SyncML compliant products is expected to be on the market early 2001.

1 The Industry Needs a Common Mobile Data Synchronization Protocol

Mobile computing has an Achilles heel – data synchronization. All the popular mobile devices – handheld computers, mobile phones, pagers, laptops – synchronize their data with network applications, desktop calendars, and other locations where information is stored. This ability to access and update information on the fly is key to the pervasive nature of mobile computing. Yet, today, almost every device uses a different technology for performing data synchronization.

What is a Synchronization Protocol?

By definition, mobile users are not always connected to a network and its stored data. Users retrieve data from the network and store it on the mobile device, where they access and manipulate the local copy of the data. Periodically, users reconnect with the network to send any local changes back to the networked data repository. Users also have the opportunity to learn about updates made to the networked data while the device was disconnected. Occasionally, they need to resolve conflicts among the updates made to the networked data. This reconciliation operation – where updates are exchanged and conflicts are resolved – is known as data synchronization.

Data synchronization, then, is the process of making two sets of data look identical. For a mobile device, synchronization applies to the data that the mobile device stores locally.

A data synchronization protocol defines the workflow for communication during a data synchronization session when the mobile device is connected to the network. The protocol must support naming and identification of records, common protocol commands to synchronize local and network data, and it can support identification and resolution of synchronization conflicts.

The Synchronization Problem Today

Today the industry offers a variety of non-interoperable data synchronization products, each connecting data from a few types of data repositories to a few devices. Each protocol functions only for selected transports, and is implemented on a few devices. Most synchronization products use different communication protocols over the network.

The proliferation of non-interoperable synchronization technologies complicates the tasks of users, device manufacturers, service providers, and application developers. The lack of a common data synchronization protocol is impeding the growth in use of mobile devices, restricting users' ability to access data, and limiting the delivery of mobile data services.

Benefits of a Common Synchronization Protocol

Let's look at each group that will benefit from adoption of an industry-wide data synchronization protocol.

- **End Users** – Today's user of mobile devices is probably using a different synchronization product with every device. That is, there is one procedure for synchronizing files between a laptop and networked data, another for synchronizing calendar on a handheld computer, and yet another remote access to email. Each technology can synchronize only a few applications, or is limited to a particular type of network connection. This arrangement is expensive to install, confusing to configure and operate, and costly to administer. With SyncML, they will be able to buy devices that synchronize with a broader range of data.

- **Device Manufacturers** – While every device manufacturer wants to support the technologies that will support the data access needs of all users and service providers, in practice a device will support one data synchronization technology. This choice is forced upon the manufacturer by constraints of storage space, memory, power consumption, and cost. Device manufacturers will benefit from a common protocol that will make the device interoperable with a broader range of applications, services, and network and transmission technologies.

- **Service Providers** – Service providers moving into the growth arena of application hosting are particularly concerned that a proliferation of synchronization technologies will make it impossible to deploy and support their customers in a cost-effective manner. Already, to support the range of data types and devices in use, service providers must install and configure multiple server infrastructures, maintain and support that infrastructure, and maintain compatibility and performance. The alternative now available, to use a single solution for data connectivity, involves the risk of a tight coupling to a proprietary solution. With SyncML, they will be able to provide connectivity to a wider selection of applications.

- **Application Developers** – Choosing to support multiple synchronization technologies enables an application to support more types of devices and networked data, but that choice comes at a cost. The developer loses the flexibility of evolving the choice of networked data repository while maintaining backward compatibility. It also increases the cost of application development and the complexity of the resulting product. The added complexity of the networked data repository can create a barrier to installation and adoption by service providers. With SyncML, they will be able to develop an application that can connect to a more diverse set of devices and networked data.

2 Characteristics of a Common Synchronization Protocol

The goal of a common synchronization protocol is symmetric. It would connect any to any, over any network. That is, it would:

- Synchronize networked data with any mobile device
- Synchronize a mobile device with any networked data

The data synchronization protocol would synchronize networked data with many different devices, including handheld computers, mobile phones, automotive computers, and desktop PCs. A user could access and manipulate the same set of data from different devices. For example, a user could read e-mail from either a handheld or a mobile phone, and still maintain a consistent, updated record of which messages had been read.

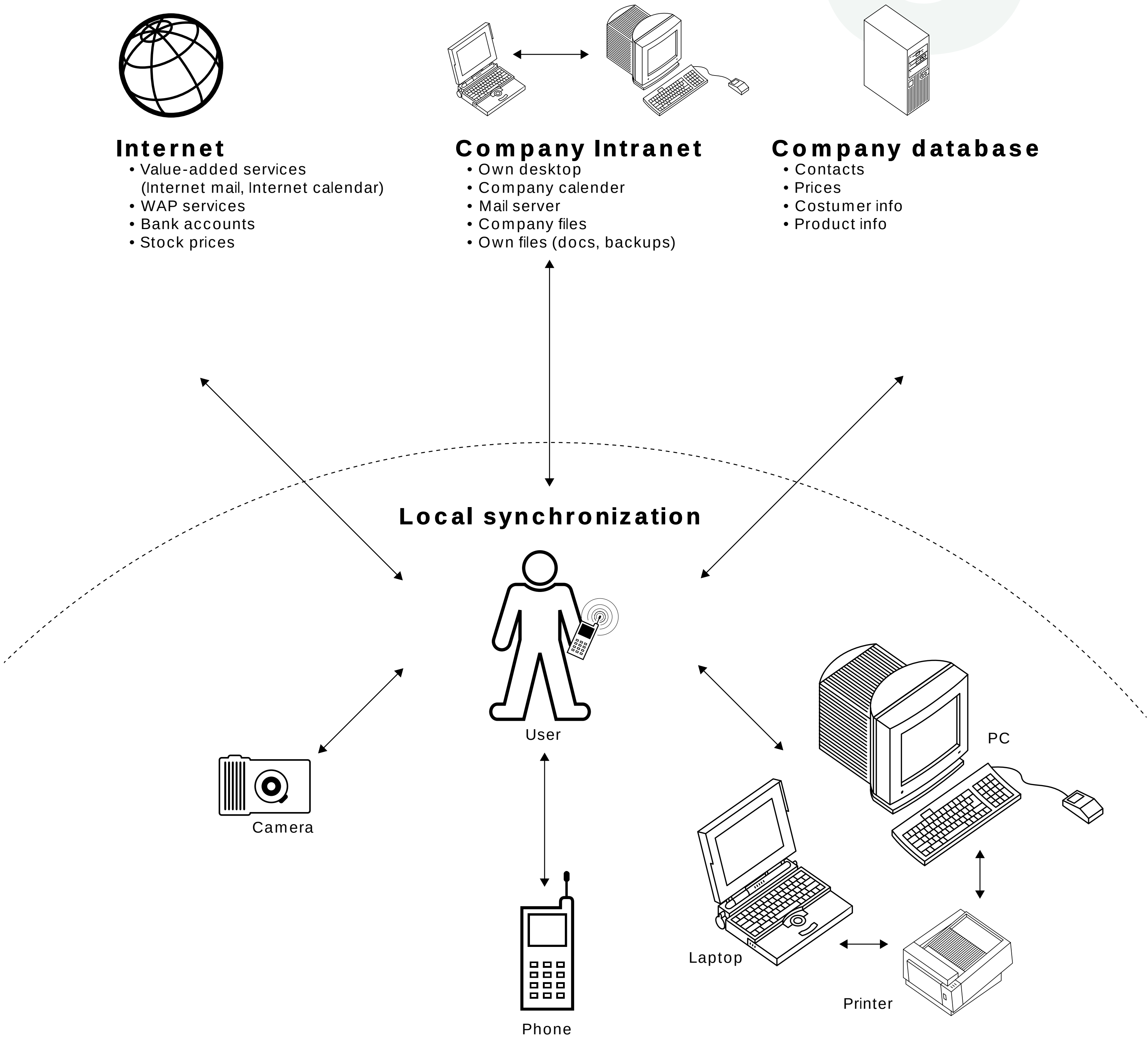
Similarly, with any-to-any synchronization, mobile devices could support more types of data, including e-mail, calendar, contact management information, enterprise data stored in databases, and documents on the web. With such functionality a user who received an order via e-mail could access the company inventory system on the same device to determine a delivery date.

To accomplish this goal, the protocol needs the following characteristics:

- Operate effectively over wireless and wireline networks
- Support a variety of transport protocols
- Support arbitrary networked data
- Enable data access from a variety of applications
- Address the resource limitations of the mobile device
- Build upon existing Internet and Web technologies
- The protocol's minimal function needs to deliver the most commonly required synchronization capability across the entire range of devices.

Scenarios for synchronization

Remote synchronization



Internet

- Value-added services (Internet mail, Internet calendar)
- WAP services
- Bank accounts
- Stock prices

Company Intranet

- Own desktop
- Company calender
- Mail server
- Company files
- Own files (docs, backups)

Company database

- Contacts
- Prices
- Costumer info
- Product info

Local synchronization

Effective Over Wireless and Wireline Networks

A common synchronization protocol must work over all networks used by mobile devices, both wireless and wireline. The various wireless networks commonly used by mobile devices demand the most from a protocol, since these wireless networks share common features of high network latency, limited bandwidth, relatively high packet costs, and low reliability of both data and connectivity.

- **High network latency** – Network latency is the delay introduced when a packet is momentarily stored, analyzed and then forwarded. Wireless networks, with a high latency, require a robust synchronization protocol.

- **Limited bandwidth** – To minimize bandwidth requirements and the associated processing demands, the protocol should allow alternate binary encoding techniques to both the data and the synchronization commands. The WBXML (WAP Binary XML) standard adopted by the WAP Forum and submitted to the W3C represents a good candidate encoding technique for limited bandwidth environments.

- **Relatively high packet costs** – The protocol must minimize the number of request-response interactions between the device and the networked data. An optimal protocol would generate a single request-response pair of messages. In this model, a request from a mobile device would contain all updates to its local data. The response message would provide the updates, with conflicts already identified and possibly resolved. Any protocol adopted by the industry should seek to enable this minimalist messaging model.

- **Low reliability of both data and connectivity** – In order to function with only intermittent connection to the network, the protocol must survive inadvertent disconnection during the synchronization procedure. Even when a disconnection is encountered, the protocol must ensure that the device and the networked data repository stay consistent and make progress when the connection is reestablished.

Support Various Transport Protocols and Media

Since wireless networks employ different transport protocols and media, a protocol must work smoothly and efficiently over:

- HTTP[1] (i.e. the Internet)
- WSP (the Wireless Session Protocol, part of the WAP protocol suite)
- OBEX (i.e. Bluetooth, IrDA, and other local connectivity)

It can also be deployed over:

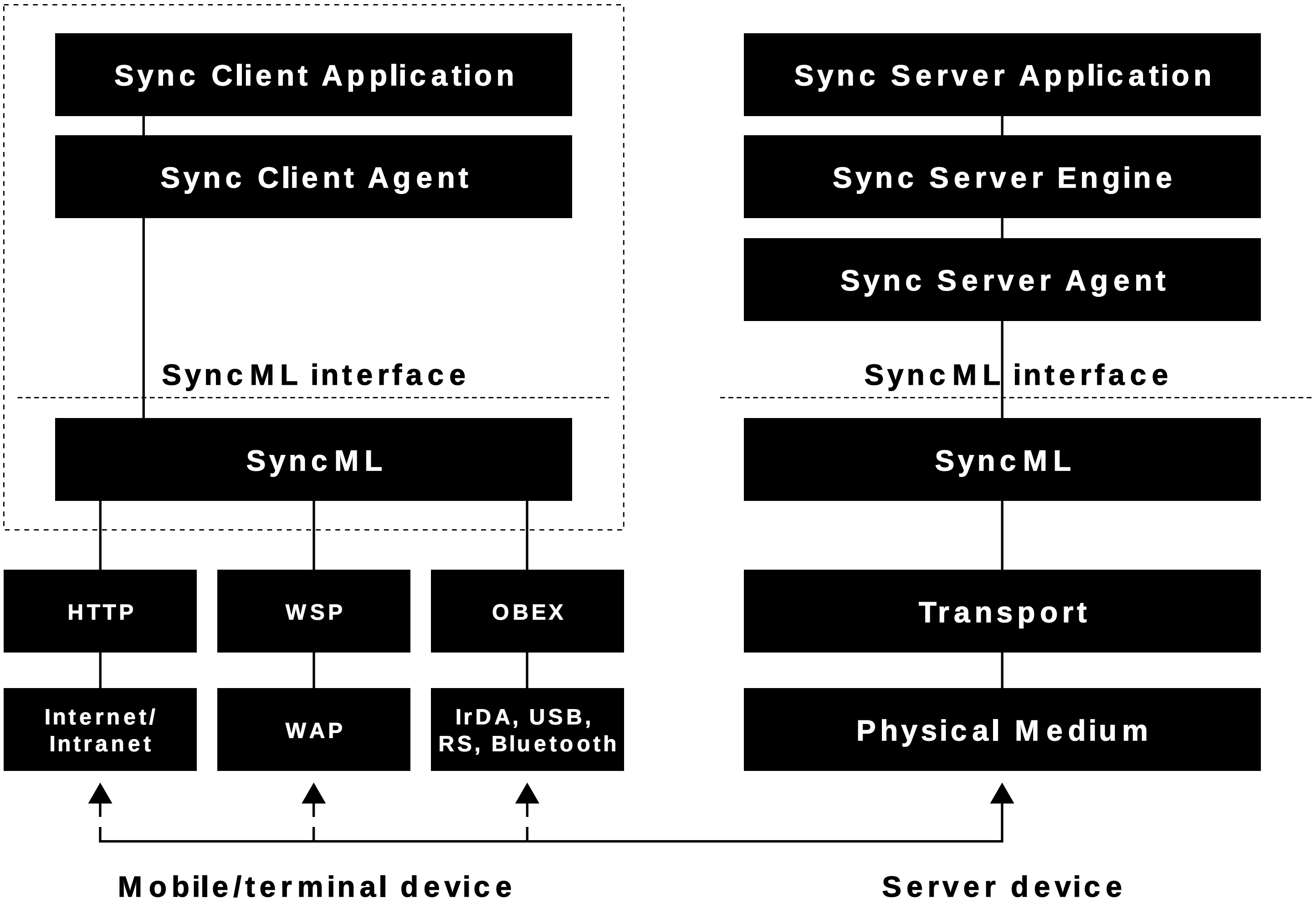
- SMTP, POP3, and IMAP
- Pure TCP/IP networks
- Proprietary wireless communication protocols

An effective synchronization protocol cannot depend on any capabilities that cannot be made available over these transports. To be efficient, the protocol should minimize duplicating features provided by the transport.

A reliable request-response model can be efficiently deployed across all of these transport protocols. An effective protocol could be built on this model. Moreover, defining a MIME (Multi-Purpose Internet Mail Extensions) content-type for the protocol units will allow the protocol to be transported across any of these different transports. Optional enhancements to the transport protocol should include security and compression capabilities.

[1] Strictly speaking, HTTP usually runs over TCP/IP, so one might simply rely on the underlying TCP/IP connectivity. However, in many virtual private network (VPN) environments, HTTP is the only protocol that is allowed through corporate firewalls. Moreover, the HTTP abstraction has been deployed over a variety of proprietary networks and is, therefore, a useful abstraction in its own right.

SyncML Protocol Architecture



Support Arbitrary Networked Data

Since a design goal is for mobile devices to communicate with a broad range of networked data, the protocol must support concurrent synchronization with multiple network data repositories. The protocol should not mandate how data is represented or structured on the device or within the networked data repository after synchronization is complete.

To ensure interoperability, the protocol must describe how common data formats are represented over the network. To ensure extensibility, the protocol should permit the definition of new data formats as needs arise. In addition, the protocol must allow implementers to define experimental, non-standard synchronization protocol primitives.

The common data formats that a protocol must support from the outset include:

- Common personal data formats, such as vCard for contact information, vCalendar and iCalendar for calendar, todo, and journal information

- Collaborative objects such as e-mail and network news

- Relational data

- XML (the Extensible Markup Language) and HTML documents

- Binary data, binary large objects, or “blobs”

Enable Data Access from a Variety of Applications

The specification shall be programming language independent. In order to work effectively with many applications, the synchronization protocol must not make assumptions about the programming language and cannot assume that both ends of the synchronization process share a single language environment.

To facilitate rapid deployment of the protocol the reference code shall be provided for a common programming language. However, this will not restrict any implementation to only this language.

Address the Resource Limitations of the Mobile Device

Mobile devices have limited memory and processor capacity, a characteristic that sets the rules for developing a synchronization protocol.

The protocol implementation needs to fit within the memory footprint of the common mobile devices on the market today, in either static code or run-time execution space. In addition, the data exchanged by the protocol itself should be small and require minimal code to transfer it to and from the device. Data exchanged using the protocol may be encoded in a binary format (such as WBXML) to reduce memory requirements for storing received synchronization messages and reduce processor resources needed to parse and process that data.

A small footprint and processor requirement gives the protocol several key advantages. A small code size eases the porting overhead and increases the likelihood that implementations will be available for all processor and/or operating system platforms. It also reduces device manufacturers' cost of adoption, an important consideration.

Optimally, a subset of the protocol would support embedded systems specialized for a particular type of data.

Build Upon Existing Internet and Web Technologies

Where possible, the protocol needs to make use of existing Internet and Web technologies. These technologies are implemented widely and are well tested, so their use within the protocol will ensure easy implementation and interoperability testing.

XML, the Extensible Markup Language, is rapidly emerging as the *lingua franca* for representing structured data over the Web. To the extent possible, the protocol should use XML to represent data being exchanged during a synchronization session.

Other useful standards in this space include MIME for registering the format of the data synchronization protocol messages.

Promote Interoperability

The protocol needs to be interoperable, synchronizing networked data with any mobile device and synchronizing a mobile device with any networked data. This includes a market space where mobile devices are small-footprint devices with minimal processor and memory resources and powerful network servers capable of executing sophisticated synchronization logic.

3 The SyncML Initiative

To accelerate the market's vision of ubiquitous data access from any device to any networked data, Ericsson, IBM, Lotus, Motorola, Nokia, Palm Inc., Psion and Starfish Software have formed the SyncML initiative. This industry initiative will deliver a common synchronization protocol that can be used industry-wide.

SyncML initiative will work with end users, device manufacturers, data providers, infrastructure developers, application developers, and service providers to define a common mobile data synchronization protocol, SyncML. This protocol will meet the resource constraints of mobile devices and wireless networks and will provide the extensibility to support a range of data types. The goal of the SyncML initiative is to deliver the protocol in the future for formal adoption and maintenance by an established standards body.

To enable adoption of the SyncML, SyncML initiative will deliver:

- An architectural specification
- Two protocol specifications (SyncML representation protocol and SyncML synchronization protocol
- Bindings to common transport protocols
- Interfaces for a common programming language
- An openly available prototype implementation of the protocol

Information about the group, including information on how to join the SyncML initiative, is available on its web site www.SyncML.org

The SyncML initiative is actively recruiting members of existing groups to join in the SyncML efforts.

SyncML

Please learn more about the SyncML initiative,
follow the progress of SyncML specification or apply for
membership on **www.SyncML.org**

SyncML