RZ 3130 (# 93176) 05/17/99
Computer Science/Mathematics 10 pages

Research Report

Service Provisioning for the “Wireless Application Protocol”

Stefan G. Hild

IBM Research Division
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

LIMITED DISTRIBUTION NOTICE

This report has been submitted for publication outside of IBM and will probably be copyrighted if accepted for publication. It has been issued as a Research Report for early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies (e.g., payment of royalties.

Research Division
Almaden ● Austin ● Beijing ● Haifa ● T.J. Watson ● Tokyo ● Zurich

Service Provisioning for the "Wireless Application Protocol"

S. G. Hild
IBM Research Division, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland

Abstract **-** This paper is based on a transcript of a presentation given by the author to Airline IT representatives at the "IATA Airlines@Internet 99" conference, under the title "Reaching the Mobile Traveler with WAP – and why you want to know!". By focussing on applications that are expected to be of interest to the audience, such as customer care and provision of flight information/e-tickets to mobile users, the presentation and the present paper attempt to give a flavor of the huge potential of the "Wireless Application Protocol (WAP)" to deliver content to customers via a personal device, where and when it matters. The paper briefly introduces WAP and its features, outlines possible scenarios for provisioning WAP service offerings, and describes their effects on any existing HTML infrastructure.
The complete WAP specifications exceed 1000 pages. This paper can of course merely touch upon the major technical aspects. For more detailed information, the reader is referred to the list of references at the end of the paper and the contact address of the author.

1 Introduction

Information dissemination has long been a focus of the computer and IT industry, and has probably reached its ultimate realization in the form of the World-Wide Web. Arguably, the pillars of the success of the Web have been its global reach – there are few countries left on the planet that are not represented within the network – and its universally accepted data representation (HTML) [1, 2] and communications (TCP/IP) standards [3].

However, the Web is not reaching everybody: comparatively sophisticated computer equipment is needed, which is expensive both to obtain as well as to maintain, and sometimes difficult to use. There are, and have been for some time, efforts to enable simple and affordable access to the Web, most of which have centered around Web-enabling other consumer devices such as telephones ("smartphones") or television sets. Another user group that has so far been "Web-less" are mobile users: Although the capability and capacity to deliver digital information to mobile devices became available with the first cellular mobile phones, pricing, speed, and reliability of such mobile data connections have been such that this technology has enjoyed only limited success in the marketplace. With the addition of digital mobile communication systems (such as GSM and CDMA) speed and reliability have improved to the extent that running the popular Internet data communication protocols (that provide the foundation for World-Wide Web services) to mobile handsets is now becoming an option. Unfortunately, World-Wide Web content is becoming increasingly complex and requires more and more bandwidth. Mobile devices, on the other hand, feature small user interfaces, i.e. a small display area and only a few input keys (a numeric key pad). Delivering content to such devices is the goal of the Wireless Application Protocol (WAP).

2 WAP Background

The WAP Forum is a 100-company-strong industry consortium that is set to enable small mobile devices to access internet style content [4]. The list of consortium member companies is impressive and includes device manufactures (in particular handset manufacturers such as Nokia, Ericsson, and Motorola, but also PDA producers such as Psion), network operators (such as AT&T, Bell South, and most European telecommunication operators), and classical IT companies (such as IBM, among others). The consortium has also received backing from well-established and respected standardization bodies such as the European Telecommunications Standardization Institute (ETSI), which is responsible for the development of the GSM digital mobile telephone system, and W3C, the consortium that oversees the development of World-Wide Web technology.

Version 1.0 of the WAP specification was released to the public in April of '98 [5]; version 1.1 is expected to be released around April of '99. The standard is comprehensive and includes specifications for the data transport levels as well as the application levels (see Figure 1).

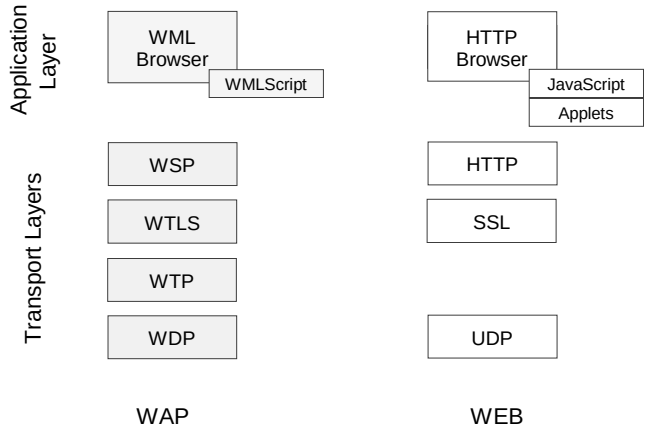

Figure 1: WAP's transport layers (WDP: Wireless Datagram Protocol; WTP: Wireless Transaction Protocol; WTLS: Wireless Transport Layer Security; WSP: WAP Session Protocol) and application layers (Browser and scripting environment), and their close Internet equivalents.

2.1 WAP Transport Layers

WAP's transport layers have been optimized for mobile data links with their characteristic throughput and bit error rates. The transport layers closely correspond to the OSI networking reference model and ubiquitous Internet standards; deviations have been made where they lead to significant performance improvements.

In detail, the WAP transport layers include the Wireless Datagram Protocol (WDP), which corresponds to the Internet's UDP protocol [6]. Where the underlying transport service is IP-based, WDP is in fact functionally as well as behaviorally identical to UDP. Building on top of WDP is WTP, the Wireless Transaction Protocol, which provides a reliable request-reply mechanism on top of the unreliable UDP/WDP bearer layer [7]. Yet on top of WTP is WTLS, WAP's security layer that corresponds to the Internet's TLS security system and is capable of providing secure transmission between the WTLS end-points [8]. Finally, the Wireless Session Protocol (WSP) provides the top layer of the transport stack and is functionally equivalent to the Internet's HTTP [9, 10].

2.2 WAP Application Layers

Like the Web, WAP is browser-based, meaning that the client's application layers are formed around a standardized browser that displays content received in a standard format. Using the browser approach allows the user to switch between devices without having to readjust to the new user interface, and provides device independence for the application service providers. WAP has defined the Wireless Markup Language (WML) as the WAP-equivalent to the Web's HTML [11]; WML is based on the upcoming XML standard that will also form the basis for the next generation of HTML. Thus, WML and HTML are expected to converge over time (more details on this issue will be presented in Section 4.3). The WAP application layer also includes a script interpreter that may be used to adjust and generate content dynamically on the end-user device [12].

3 Why Provide WAP Services?

WAP service provisioning is attractive for both the network operator as well as third parties. For the network operator, value-add services that are quickly and device-independently developed

and deployed using WAP as a delivery methodology are valuable tools for network differentiation and customer retention. Customer "churn" has been a serious problem for some network operators, in particular in Europe, where competition between networks is especially fierce and where up to 25% of the user population is constantly migrating between seemingly equivalent networks to take advantage of small price cuts. At the same time, competition has led to price and tariff erosion that, over the past 5 years, have led to a continuous decline in revenue per device [13], although some of this decline was compensated by the huge increase in device numbers over the same period of time.

Service providers, on the other hand, can provide WAP services with comparatively a small investment and still reach a large user community: It is commonly expected that by 2002 more than 250 million cellular handsets will have been sold worldwide [14]. The WAP Forum's member list features many of the most active handset developers and manufacturers, in particular, Nokia, Ericsson and Motorola, which together command about 80% of the world's handset production. It seems therefore reasonable to expect that many of those devices will incorporate WAP functionality. Furthermore, cellular phone ownership is particularly high among high-income households and WAP therefore provides an interesting medium for corporate communications.

Finally and arguably most importantly, cellular phones are essentially "personalized" devices and can potentially deliver the client identity as well as the client profile and geographical location to service providers. It is therefore possible to provide WAP services that are tailored to the particular user in his/her current location. Being a mobile "companion device", users tend to develop a special trust relationship with their PDAs – more than desktop computers or corporate mainframe systems, these devices are entrusted with personal address lists or schedule information. It therefore seems likely that users will also be more willing to utilize their PDA when making electronic purchases.

As an example of an application that illustrates the use of WAP as a corporate communications platform, consider the purchase of an electronic airline ticket (e-ticket) with a WAP-enabled handset (Figure 2 illustrates a demo-example based on a WAP simulation environment developed at IBM Zurich which accesses the Easy Sabre HTML interface and was developed as our first WAP demo service in early '98 [15]).

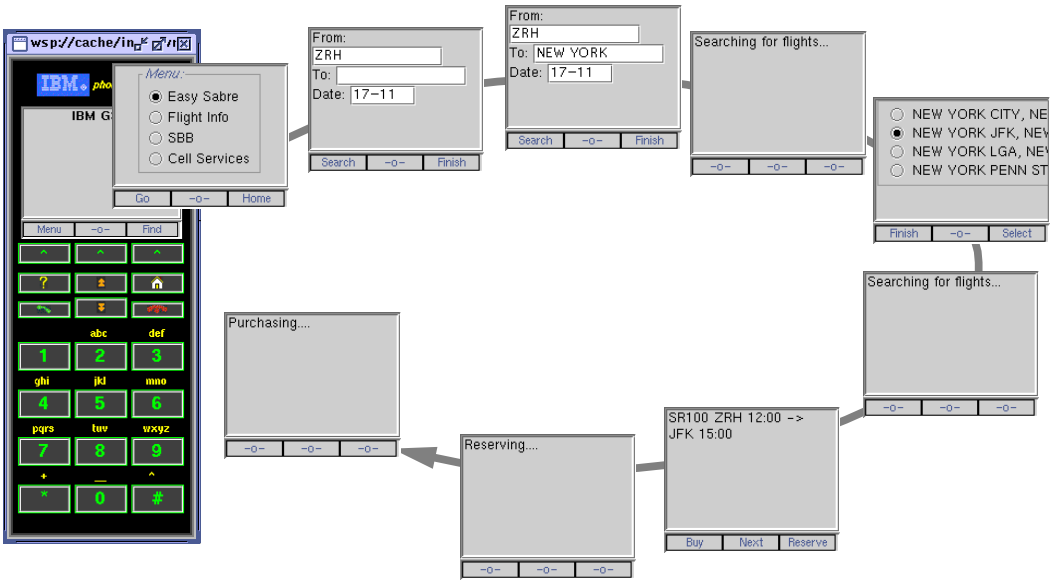

Figure 2: Sample WAP application: An e-ticket purchase.

The screen shots display the sequence of events necessary to select a flight itinerary from Sabre's databases to order the e-ticket. Once completed, the ticket is stored in the user's handset and, provided the airport is equipped accordingly, one might imagine that the check-in procedure could be completed by transmitting the e-ticket from the handset to the check-in terminal (using the phone's local infrared communication's channel).
Using WAP's built-in push functionality, the airline may choose to provide status information to the user. In case of a delay, the application provider can push an appropriate information panel to the user device and have it displayed to the customer. One key benefit of WAP over existing pager technology is that every delivery of information is in effect the delivery of an application that allows the service provider to enter into a new interaction with the user. For example, a notification of a flight delay or cancellation may be combined with an offer to rebook the passenger onto another flight or to arrange a hotel booking. The user can choose the option he/she desires and inform the service provider with the click of a button.
As with all WAP applications, location information that may be provided from the network operator can be used to customized or trigger the delivery of content to the user. For example, passengers arriving in the terminal building can be informed of the departure gate number.

4 Moving from the Web to WAP

Although convergence is expected with regard to the markup languages used by the Web and WAP to the extent that both will be based on the Extensible Markup Language (XML), it is unlikely that existing HTML-based Web offerings can be accessed by most WAP devices without modification. By and large, this is due to usability reasons rather than technical problems: Existing Web offerings were designed for the desktop user, who can access a large and powerful computer with fast Internet connection. Such desktop devices are capable of retrieving and rendering graphically rich and complex content. Although there are technical solutions that reduce the transmitted information to suit the much slower speed of mobile data connections and the limited display capabilities of mobile devices (such solutions are being employed by systems such as "compact HTML" [16]), usability suffers as user interfaces and interactions that had been designed for a stationary user in front of a large display translate poorly to mobile users. In general, application adaptation for WAP involves two steps:

- The careful reduction of the displayed content to provide a readable and suitably "dense" user interface.
- The redesign of the application interaction and user guidance to minimize the number of screens/keystrokes necessary to complete a transaction.

As an example, Figure 3 compares the top-level page of Sabre's Web interface with the top-level input panel of our WAP-based flight reservation system (which is based on the Sabre Web offering). The WAP interface can reduce the number of required user inputs based on client identity (which we use to execute the login procedure to Sabre), and the location information delivered from the cellular network operator (which we use to guess the user's departure airport).
WAP offerings can easily be combined with their existing Web counterparts; application training is better provided on the Web than via WAP. Unlike an anonymous Web user, WAP delivers user-specific information such as user id, user profile, and location, that can be used to personalize the delivered content.

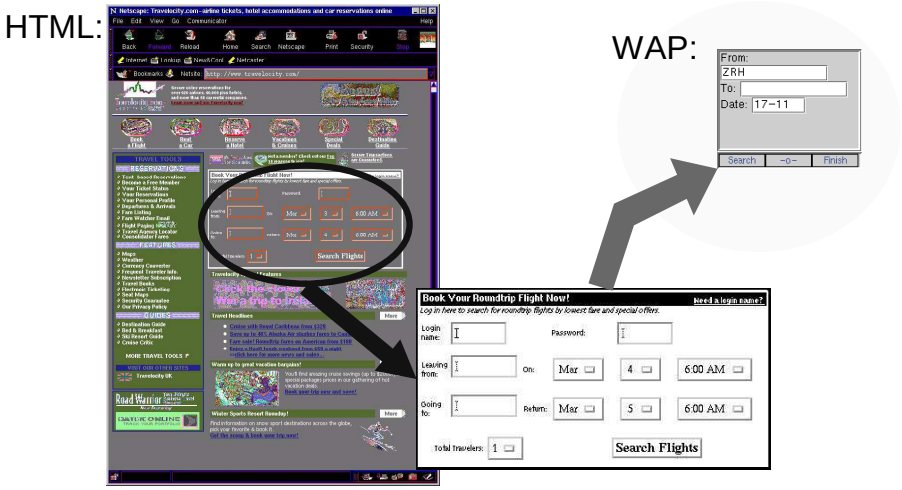

Figure 3: Transition from a Web-based interface to the WAP interface: Reduction of information by focusing on the necessary and utilizing network-delivered environment information.

4.1 WAP Backend Infrastructure

Figure 4 illustrates a typical WAP backend infrastructure. On the left-hand side, the WAP browser is shown on the client device. Similar to the World-Wide Web, the browser retrieves WAP content from the Origin Server that is shown on the right-hand side of Figure 4. Also shown in Figure 4 are other functional elements that are expected to form part of a typical WAP delivery chain: the WAP gateway bridges the gap between the mobile data network and the wired link to the Origin Server. Front-end machines are used to generate a device- and user-specific view of the origin-server's data structures and service offerings.

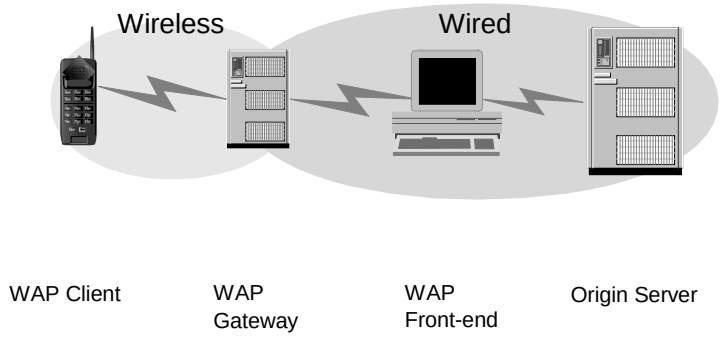

Figure 4: Possible WAP content delivery chain from WAP Server to WAP client.

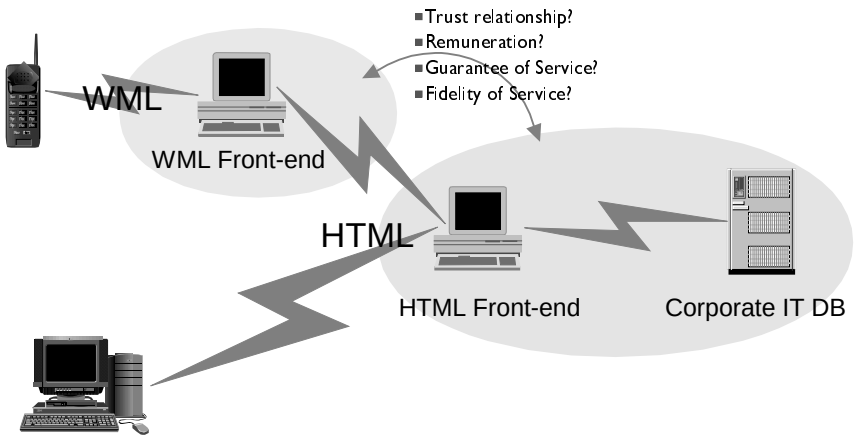

Figure 5: Daisy-chaining front-ends to support WAP devices from existing HTML offerings.

The addition of WAP offerings to existing Web services within the Origin Server is illustrated in Figure 5. Here, a conventional setup is assumed that uses an HTML front-end to generate an HTML-based view of the corporate IT infrastructure. Initial WAP offerings may be based on the same HTML views: An additional WML front-end is daisy-chained to the existing HTML offering to generate a separate WML view. This solution is quick to deploy because the WAP-related components are completely shielded from the corporate IT infrastructure; the WML front-end machine is simply a client to the HTML front-end. Front-ends are implemented quickly using platform solutions that can be readily obtained from IT vendors. A platform that is especially geared towards WAP with easy integration with HTML as well as other data sources and convenient generation of WML traffic is currently the subject of work at IBM Zurich Research.

Such solutions can be deployed quickly as the running of a WML front-end can potentially be out-sourced to third-party service providers, a solution that should be of particular interest to small application providers that wish to make only a small investment to ground-test the WAP service acceptance of their clients. Larger organizations may require strong service guarantees that may not be available from such third parties.

The security implications of providing potentially security-sensitive applications via untrusted third parties are covered later in this paper.

Although quick to deploy, any WAP service offered by daisy-chaining such front-end platforms onto existing HTML offerings is naturally limited to the facilities and features provided by the HTML offering. Such a solution is therefore necessarily unable to utilize many of WAP's most interesting features, in particular data push and the ability to delivery WAP-customized data (e.g. by use of WAP-delivered location information).

These advanced facilities require that the WAP service offerings be tied deeper into the corporate IT infrastructure. Figure 6 illustrates such a setup; here, a separate WML front-end duplicates the existing HTML setup and is provided exclusively to generate a WML service offering. This level of integration, however, is likely to require investment levels that are equivalent to those required to provide the first Web offerings. What is even more significant is that ongoing maintenance expenses are incurred by the WML service that, owing to the lack of integration with the HTML offering, must be maintained separately. This is likely to lead to out-of-sync service roll-out in the HTML space and the WML space, even where services are concerned that can be supported equally well in both domains.

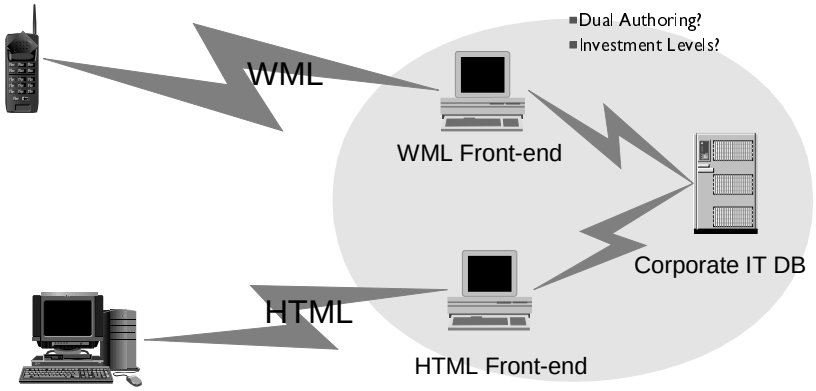

Figure 6: Separate front-ends for the HTML and WAP offerings allow closer integration into the corporate IT infrastructure and enable the service provider to make full use of WAP's potential, but require dual authoring and double investments.

4.2 XML

As indicated above, it is apparent that one of the reasons for the success of the World-Wide Web is the availability of a platform-independent data representation format, namely HTML. HTML allows service providers to generate content that can be accessed and viewed (i.e. browsed) on a large number of client computers, with the only common requirement being the support of an HTML browser. Although HTML itself started out as a simple markup language, it grew over the years to accommodate the desire from users and application providers to supply more complex content, such as multimedia and improved interactivity. This development was supported by the availability of ample processing power and adequate communications bandwidth in even the simplest of desktop client devices. To the same extent that HTML adjusted to the needs of desktop users and embraced the capabilities of the desktop paradigm, it also became increasingly bound to that paradigm. As a result, it is now difficult to adapt desktop HTML for devices that offer less processing power than, say, an Intel 486 platform, and less communications bandwidth than, say, a 28.8 kbps connection. Attempts to achieve this do exist, but mostly rely on simply subsetting the HTML standard, and on author cooperation to comply with HTML usage recommendations rather than technical vigor. Neither of these two methods are universally accepted; the first removes functionality from the standard without always offering alternatives, whereas the second cannot be enforced and leaves much room to interpretation by individual content authors.

Among HTML's many failings, the one that is responsible for this development is its lack of separation between *data* and *data representation*. HTML pages contain both, i.e. data content *and* instructions on how that content is to be presented to the user. Whereas the content itself is neutral to changes in the user's environment such as the switch to a different client device, the representation of said content may need to be varied, depending for example on the screen space available at any given end-user device. Hence, the breadth of available devices with different user input and output capabilities and different connectivity paradigms appears to demand an equally broad range of variations of a single HTML document. However, this would lead to a fragmentation of the World-Wide Web, which in turn would either limit the reach of any service offering in any particular markup language, or would pressure service providers to author content in many different markup languages (for first signs of such fragmentation see [17]).

This dilemma has been recognized by the W3C, the World-Wide-Web Consortium [18], and work is under way to base the next generation of HTML on XML, the Extensible Markup Language [19, 20]. XML itself is a syntax definition language that allows content sources to define data sets and formats, and to encode their data in a universally accepted format. In addition

to the raw data stream that the XML document represents, the data provider also sends rendering instructions separately from the data stream. A separate standard by the name of XSL (Extensible Style-Sheet Language) has been proposed for this purpose [21]. Both streams together form input to an XML-capable browser on the client device. The interpretation of the provided XSL document may take into account the actual device characteristics; this task can be done either at the application provider, the device itself, or at an intermediate transcoder or view generator (see Figure 7).

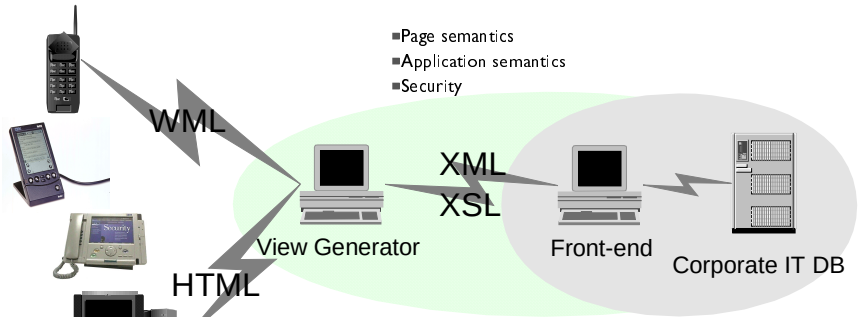

Figure 7: Provision of device-independent services based on XML and XSL-interpreting view generators.

Such a setup has a number of benefits to both the application provider as well as the user and device manufacturers. Provided sufficient information is contained within the XSL document, the application provider may be completely shielded from the devices behind a third-party view generator. New devices are accommodated by view generators that can be provided by the device manufactures; to view an application provider's content on such a device, the content traffic is merely routed via an appropriate view generator.

4.3 Security Implications

WAP's security layer WTLS (WAP Transport Layer Security) provides encryption/decryption services and a secure link between the end points of the WAP communication layers. However, those end points may not always be located at the actual end points of a transaction. In particular, if third party transcoder services are used then those transcoders require plain-text access to the communicated data traffic in order to provide effective device adaptation services. A secure transport layer may still be used to secure the connections between the service provider and the third party transcoder, and from the transcoder to the device, but for many security-sensitive applications the requirement to disclose plain text to any third party within such a hop-by-hop secured link is prohibitive. Entering into trust relationships with a third-party transcoding service may alleviate this problem to some extent but it seriously limits one's ability to utilize a broad range of value-add transcoder services. IBM Zurich Research has contributed the so-called SerVANT technology to WAP (see Figure 8) [22, 23]. SerVANT enables third-party transcoding while maintaining end-to-end security between the application service provider and the client on the security-sensitive parts of the data stream. Experience shows that only small sections of the overall stream are actually security sensitive (such as a bank balance within an otherwise non-security-sensitive boiler-plate); restricting the end-to-end encryption to those parts saves processing resources that would otherwise be spent on encrypting and decrypting non-security-sensitive information and allows the disclosure of those sections to third parties. This non-secure stream provides sufficient information for transcoders to complete their tasks.

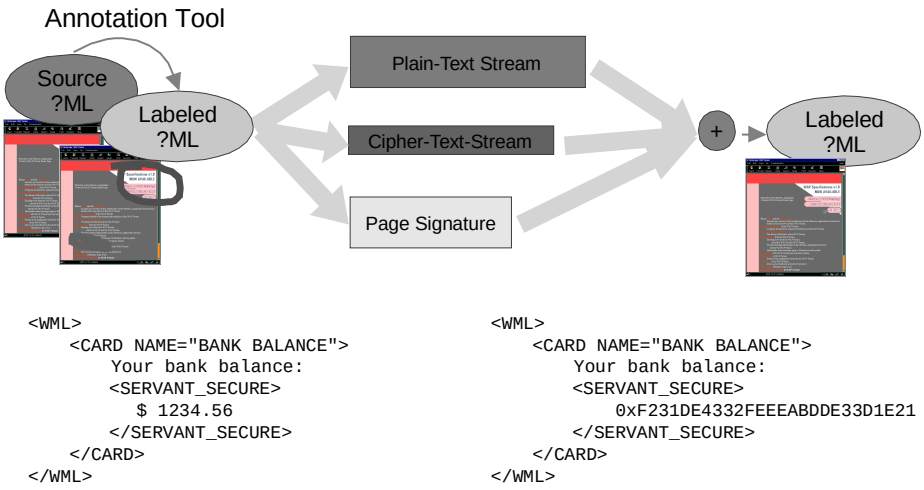

Figure 8: 'SerVANT' enables secure and verifiable transcoding within a delivery chain that incorporates content-manipulating third party transcoders.

5 Conclusions

WAP, the Wireless Application Protocol is an upcoming standard for enabling mobile devices with limited input and output capabilities to access application servers and content providers similar to the way desktop computers access Web pages. WAP is driven by a large industry consortium with buy-in from companies that represent a large and powerful majority of handset manufacturers, in addition to many participants from the IT industry and major telecommunications operators. For many of those consortium members, "1999 is the year of WAP!". First devices have been announced and will become generally available in the course of 1999.

The potential benefits for application and service providers are the ability to reach customers through a personal device and to utilize additional user information (user id, profile, location) to delivery information that matters when it matters. Investments are necessary to develop new WAP service offerings or to enable existing HTML service offerings for WAP. This paper has touched briefly upon possible service provision scenarios and possible convergence paths to provide a uniform service/application offering to a growing range of devices, of which HTML-enabled desktop computers and WAP-enabled cellular telephones are merely the first breed.

References

[1] T. Berners-Lee, "HyperText Markup Language (HTMLS)", ftp://info.cern.ch/pub/www/docs/http-spec.ps, March '93

[2] D. Raggett, A. LeHors, I. Jacobs, "HTML 4.0 Specification", http://www.w3.org/TR/RFC-html40, April '98

[3] J. B. Postel, "Transmission Control Protocol", RFC793, Network Information Center, SRI Internation, September '81

[4] The WAP Forum's homepage can be found at http://www.wapforum.org and contains details on WAP's ongoing standardization efforts as well as a range of introductory papers on the topic of WAP

[5] WAP Forum, “WAP Specification Version 1.0”, April ’98, http://www.wapforum.org/docs/technical.htm
[6] WAP Forum, “WAP WDP – Wireless Datagram Protocol Specification”, April ‘98
[7] WAP Forum, “WAP WTP – Wireless Transaction Protocol Specification”, April ‘98
[8] WAP Forum, “WAP WTLS – Wireless Transport Layer Security Specification”, April ‘98
[9] WAP Forum, “WAP WSP – Wireless Session Protocol Specification”, April ‘98
[10] T. Berners-Lee, “Hypertext Transfer Protocol (HTTP)”, Internet Draft draft-ietf-iiir-http-00.txt, IETF, November ‘93
[11] WAP Forum, “WAP WML – Wireless Markup Language”, April ‘98
[12] WAP Forum, “WAP WMLScript – WMLScript Language Specification”, April ‘98
[13] In their report “US Wireless Pricing 97-98”, the Strategis Group reports that in Europe average monthly revenue per subscriber has dropped from $324 in 1984 to $74 in 1997
[14] Source: IDC
[15] C. Binding, S. Hild, M. Moser, “Wapsody: WAP Simulation and Development Environment”, IBM Zurich Research, November ‘98
[16] T. Karnada, “Compact HTML for Small Information Appliances”, http://www.w3.org/TR/1998/NOTE-compactHTML-19980209, February ‘98
[17] T. Wugofski, “A Modular Hypertext Markup Language for Broadcast Applications”, Over the Moon Productions, http://toocan.philabs.research.philips.com/misc/atsc/bhtml/bhtml_4.html, October ‘98
[18] The W3C homepage may be found at http://www.w3.org
[19] T. Bray, J. Paoli, C.M. Sperberg-McQueen, “Extensible Markup Language (XML) 1.0 Specification”, http://www.w3.org/TR/REC-xml, February ‘98
[20] S. Pemberton, M. Altheim, D. Austin, et al., “XHTML: TheExtensible HyperText Markup Language – A Reformulation of HTML4.0 in XML1.0”, http://www.w3.org/TR/WD-html-in-xml, March ‘99
[21] J. Clark, S. Deach (eds.), “Extensible Stylesheet Language (XSL)”, http://www.w3.org/TR/WD-xsl, December ‘98
[22] S. Hild, L. O’Connor, “SerVANT – Secure and Verifiable Transcoding With Applications to Wireless Environments”, submission to WebAhead conference, August ‘98
[23] S. Hild, L. O’Connor, C. Binding, “The ‘SerVANT’ Architecture – Implementation Notes”, WAP Forum Input Document, Fort Worth Meeting, February ‘99