

## BRINGING SUBSCRIBER STATE TO APPLICATIONS MANAGEMENT

To effectively manage the subscriber experience, Service Providers must be on top of network resources and applications as they relate to subscriber demand. This allows Service Providers to plan more successfully for the launch of new applications and technology on the network; for example, by tracking how a subscriber is using the network and what applications and services the subscriber is requesting in real time. Subscriber context includes static information related to subscriber entitlements and choices as well as historical data related to usage; however, a key aspect of subscriber context is provided through dynamic state information.

This Application Brief defines subscriber state information and describes how to use it in the most effective and profitable manner to support applications management. It discusses how subscriber state is derived and presents the challenges faced by many Service Providers today when dealing with state information.

MAXIMIZE THE EFFICIENT AND PROFITABLE USE OF SUBSCRIBER STATE INFORMATION.

This application brief will be of interest to those responsible for network operations and applications deployment in Service Provider networks. It details two viable scenarios for maximizing the efficient and profitable use of subscriber state information. Service Providers can use the information to help plan for future network evolution and employ best practices to implement new services.

### WHAT IS STATE INFORMATION?

State information identifies and defines network sessions, which are tied to subscribers. It includes:

- > Information about the **network**, such as the IP address that the subscriber is currently using and what network the subscriber is currently connected to.
- > **Device** information, such as the device being used and the version of software, which enables multimedia applications to optimize the quality of the delivered content and services based on device.
- > Information about the **subscriber**, including location, if the subscriber is roaming, and services being used or requested.
- > **Unique** state information — that is, state information that is provided by the network but not necessarily covered by current standards. This unique data can be used in many ways based on Service Provider-specific architectures.

### TURNING STATE INTO PROFIT

Offering the best user experience possible translates to lower churn and increased ARPU. Increasing business agility means accelerated time to market and lower operational and support costs. The intelligent use of state information makes both of these goals possible. With a comprehensive real-time view of subscriber context, Service Providers can extend that awareness out to applications to ensure a high-quality user experience, gain better insight into user trends, and leverage a comprehensive view of the subscriber for policy and access controls to drive new services.

The ability to dynamically manage subscribers' experiences while allowing them control and choice over certain aspects of the available experiences enables Service Providers to:

- > Offer new applications and content that rely on state information, and increase ARPU.
- > Create multiple services — for example, based on time of day, day of week, or events.
- > Enable network control to ensure a quality subscriber experience.
- > Provide a managed and maximized subscriber experience to reduce subscriber churn.

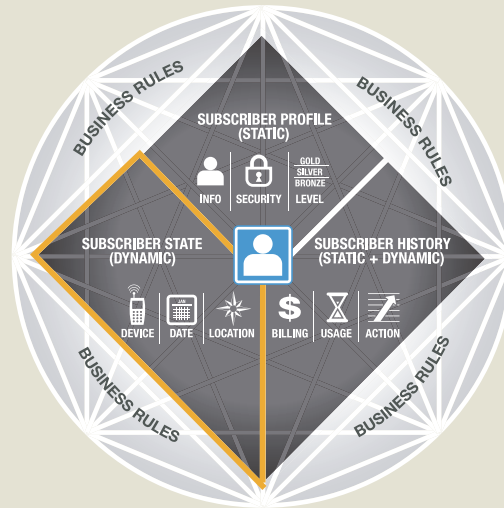
### WHAT IS SUBSCRIBER STATE?

**Determines what the subscriber is doing at a specific moment**

- > Is he on the network?
- > Which network is he using?
- > What is he doing — which application?
- > Which device?
- > Is he roaming?
- > Time of day, day of week data

#### Business Value

- > Enrich the application experience
- > Create new services based on subscriber state
- > Prevent fraud



With comprehensive context of how, where, and with what applications subscribers are using the network in real time, Service Providers can improve control and management of the overall subscriber experience. As a result Service Providers can increase subscriber retention, drive new revenue streams, and increase ARPU with innovative state-driven services, such as those detailed in table 1. To get this context, they need to be able to derive state information, and serve it to applications and content in the manner that meets the requirements of the various application and content servers in the network.

STATE-DRIVEN SERVICE EXAMPLES	
STATE INFORMATION	ENABLED SERVICES
IP Address	Application- and content-based billing.
Access Type	Access-specific multimedia content delivery (the right application, in the right format, on the right network).
Device Type	Weather, locators, mapping, regional offers, etc.
Subscriber Activity	Parental controls, historical tracking and trending, service upgrade offers, etc.
Time (of day, of week, of year)	Seasonal offers, promotional offers, birthday offers, limited-time event offers, etc. For example, free plays of online soccer game during World Cup.
IP Address, Time of Day, MSISDN	Lawful intercept.

TABLE 1: State-Driven Service Examples

## HOW IS SUBSCRIBER STATE DERIVED?

State information is typically generated either through RADIUS accounting or through interactions with the dynamic host control protocol (DHCP) server on all network types. RADIUS or Diameter accounting feeds are generated to the RADIUS engine (or the AAA server) from core network elements (e.g., packet data serving node [PDSN], gateway GPRS support node [GGSN], Access Service Network [ASN] gateway). See figure 1.

When a subscriber connects to any network and is authenticated and authorized onto the network, the core elements assign an IP address. At that time, the true session starts, and session state information is created, derived, and captured. State information provides:

- > Identity mapping — subscriber ID for the session (MSISDN, MIN, Private ID, etc.).
- > Applications chosen (PDP Context, Flow ID, APN, Flow Descriptor, etc.).
- > Network type currently used and roaming status (NAI, SGSN IP, IP Subnet, etc.).

This information comes with multiple starts, stops, and interims, each one of those changing the state information, which is updated via interim RADIUS accounting messages as the session continues.

The session ends when the subscriber ends the connection to the network. In a wireline situation that could be days or even weeks, but with a wireless connection it could be a matter of seconds.

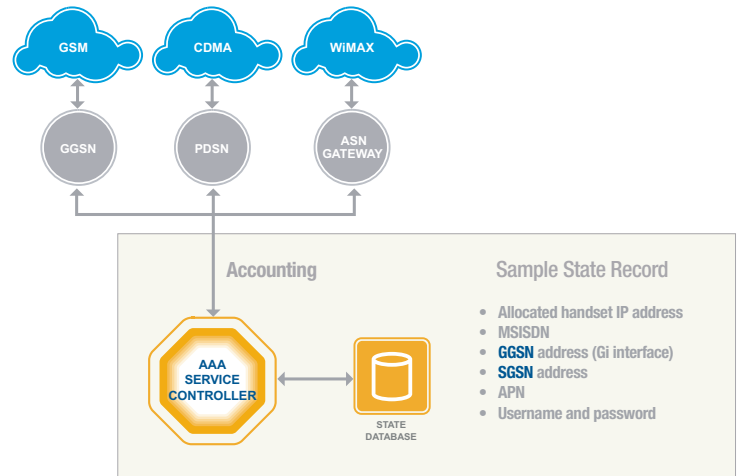


FIGURE 1: Deriving State

## WHERE STATE IS STORED TODAY

Applications and intelligent network elements (INEs) require certain pieces of information in order to enable and deliver services (e.g., subscriber identity, profiles). Service Providers typically have a subscriber database that houses this key information (the information is provisioned by the application programming interface [API] or by direct query to a common lightweight directory access protocol [LDAP] or other database).

All applications and INEs must have access to state information. When a request comes to the application or INE, the application or INE determines the identity of the subscriber based on that state information (typically IE, IP address, or other request content captured by snooping the RADIUS, Diameter, or DHCP server) and delivers the service based on subscriber profile information (taken from the internal database or central LDAP directory).

## CHALLENGES

### Too much information

Generally, application servers — including WAP gateways — that receive RADIUS accounting data do so to fulfill a need for specific identity information, usually about the subscriber related to the IP address for that session. Applications that require only the MSISDN to IP or MIN to IP address for that session, for example, can get overwhelmed by the amount of state data that can be pushed to the application server; for example, all network connection events (10,000 messages per second in some instances). Most messages are not relevant to the application server, nor is that server set up to handle a large amount of RADIUS data. An application system that tries to do this in its own memory will experience performance and crashing problems as it is forced to cache redundant data.

Application servers were not designed to scale to support this increase in non-core processing. With only a few applications in the network this can be somewhat manageable; however, as network traffic increases, application platforms must be upgraded to deal with these new and increasing performance requirements.

### In too many places

In many networks today, application servers track and store their own state data. However, there are several points in the network that require elements of this information to successfully enable a solid user experience. With state information stored in many locations, and more applications being deployed on networks, performance and provisioning issues result, introducing the potential for timeouts, revenue leakage, and the inefficiencies described in the section above.

### Adding a new application

Adding a new application or service to a network can be a time-consuming and expensive process, often requiring a new schema object for the LDAP directory that is difficult to deploy owing to testing and deployment issues on a critical centralized element. It may also require provisioning a new element on the network.

### Dual sign-on

Another of the consequences of scattered state information is dual sign-on. When a subscriber attempts to log on to the network the AAA server, responsible for authenticating the logon, has no idea if the subscriber is already logged on (unless the AAA server is state-aware). This opens the door to many problems, including fraud. In one network example, a Service Provider launched a high-speed 3.5G network in Central America, put 20 subscribers on the new network, and quickly lost all its bandwidth. The Service Provider discovered that 10 subscribers had logged on more than 5,000 times simultaneously.

### Lack of standards

Accurate and synchronized state information is vital to unified application operations in an environment supporting WAP, MMS, streaming services, and enhanced single sign-in mechanisms. Unfortunately, standards may have failed to keep up with the evolution of pre-IMS applications. While future standards may emerge from the OMA, 3GPP, or IETF, today Service Providers largely rely on significant and costly proprietary development based on LDAP or XML-over-HTTP interfaces to fill the void. However, as Service Providers add new applications and services, this approach introduces an integration challenge that drives up management costs and support and impacts time-to-market for new services.

### Value of a single repository

Many of these challenges stem from the lack of centrally stored state information. By maintaining state information centrally to serve multiple applications, Service Providers have a single repository that lets them understand what's going on in the network at any given moment. Centrally stored state information gives all application servers a single point from which to retrieve that information, eases integration of new technologies, and makes state information available to all existing servers along with any servers or applications that may be implemented in the future.

---

## TWO MODELS FOR SERVING STATE

Two models — each serving a different purpose — can be employed to effectively serve state information to applications:

- > The **push** model, which is often adopted as a migratory stepping stone to the pull model, is used for data reduction and data enrichment of session information being pushed in RADIUS streams to application servers.
- > The **pull** model provides real-time state information as part of a broad application policy solution that includes a single state repository serving multiple applications.

Ultimately, most Service Providers will want to consider the pull model as the endgame, but many networks today still use applications that aren't engineered to pull information from a central repository and so need to be pushed that information. Both models can coexist and can be complementary, and the migration path from push to pull is straightforward.

## The push model

In the push model, a common data streaming platform solution pushes session information to applications. An example:

- > Two applications are on the network: one needs MSISDN and IP address, while the other needs MSISDN, IP address, and Access Point Name (APN).
- > The push model takes the RADIUS accounting information available that may have, for instance, 14 attributes.
- > Using a data streaming solution, that single incoming RADIUS stream can be segmented into multiple outbound streams, each with information tailored to the individual destination application. The first application is returned only the MSISDN and IP address, while the second also receives the APN.

In this model the Service Provider is not creating a state repository but simply pushing the RADIUS accounting information to the applications that need it, meeting the requirements of applications that can't query a state repository but are capable of receiving the stream to get that state information.

The key components in the push model are depicted in figure 2:

- > AAA Service Controller
  - Derives state information from network elements.
  - Correlates and aggregates RADIUS accounting records from the PDSN, GGSN, or ASN Gateway and feeds this information to the Data Streaming Server (DSS).
- > Data Streaming Server
  - Provides a data streaming solution that can push session information in the form of RADIUS accounting records or Diameter data to any application.
  - Enables flexible formatting of RADIUS accounting records based on criteria (e.g., RADIUS packet type [accounting start/stop/interim]).
  - Supports the simultaneous generation of multiple accounting record formats.

These approaches are superior to current proprietary methods. All of the components described in the push and pull models are based on carrier-grade technology deployed and proven to support the most demanding tier-1 Service Provider network environments. As a result, Service Providers can be assured that the solution will scale to meet their growth needs as they deploy new services. Bridgewater Systems also has a track record with integration and provisioning to a wide range of network elements, application platforms, and OSS/BSS.

While the repository solution (pull model) is more robust and engineered to support growth and evolution of applications, the push model offers a solid and proven approach until application vendors can enable widespread support of a common state repository or database.

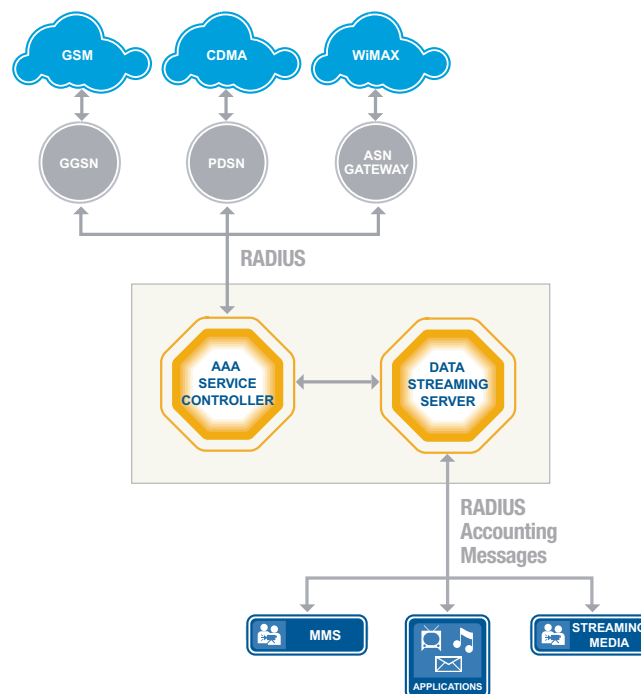


FIGURE 2: The Push Model

## The pull model

In the pull model, the accounting stream (RADIUS or Diameter) is written into an actual state database. Applications query this database to pull state data on an as-needed basis.

- > Network efficiency is increased dramatically because information that may not be relevant is not being broadcast over the network to every application server.
- > Application efficiency is increased because the application can query the database to get only the information it needs and has to query only one location. It is no longer subject to the consequences of information overload and performance challenges.
- > When a query includes a policy request, the state information can be provided with a policy response. Using policy allows for more intelligence in the way the state information is returned to the application than the straight query method. The application of business rules can, for example, provide control to how information is passed to third-party vendors. In fact, total control allows for third-party vendor "A" to get one set of information and third-party vendor "B" to get an entirely different set of information.

The key components in the pull model are depicted in figure 3:

- > Application Policy Controller
  - Offers a centralized solution for managing subscriber access to applications, enabling granular controls using policy.
  - Controls access of applications to subscriber profile data.
- > AAA Service Controller
  - Derives state information from network elements and feeds this information to the state database.
- > State Database
  - Provides a common repository in which subscriber state is maintained in real time.
  - Can be customized relative to the state data that is maintained and how often it is refreshed.
  - Is updated with state information fed from the AAA Service Controller.
  - Enables applications to "pull" state information via the Application Policy Controller.

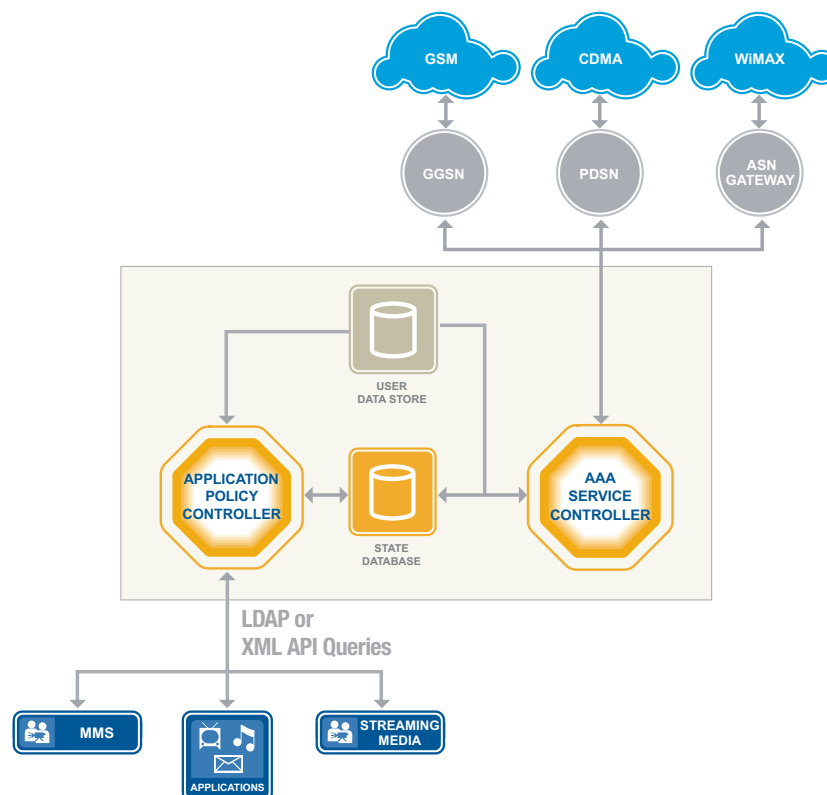


FIGURE 3: The Pull Model

## BRIDGEWATER AND THE BOTTOM LINE

Regardless of which model makes the most sense for a particular Service Provider's network, Bridgewater Systems provides industry-leading state solutions proven and deployed in tier-1 networks around the globe. Bridgewater has a solid track record of developing and providing subscriber state data that solves problems such as usage fraud, and associating IP addresses with billable identities as examples. Bridgewater offers carrier-grade technology with scalability and performance proven to support the most demanding network environments, ensuring that Service Providers can leverage a future-proof solution to meet their needs now and in the future.

Bridgewater products derive, store, and serve up state information as Service Providers need it — including seamless translation between RADIUS and Diameter information in a multiprotocol environment and a Home Subscriber Server product that provides state information in an IP Multimedia Subsystem (IMS) environment.

Centrally stored state information gives all application servers a single point from which to retrieve that information, eases integration of new technologies, and makes state information available to all existing servers along with any servers or applications that may be implemented in the future.

With state information — that is, comprehensive, dynamic context of how, where, and with what applications subscribers are using the network — Service Providers can better control and manage the subscriber experience. By providing state information efficiently and intelligently to applications, content, and services, Service Providers can improve subscriber retention with highly targeted services that ensure that the right application is delivered in the right format, on the most appropriate device, on the right network. Service Providers can drive new revenue streams and increase ARPU with a broad range of state-driven applications such as event-based applications, seasonal offers, and regional promotions.

## ABOUT BRIDGEWATER SYSTEMS

Bridgewater Systems develops the industry's most advanced subscriber-centric policy management software for fixed, mobile, and converged networks. Its solutions help global Service Providers launch new services faster and maximize profits by creating a subscriber-centric policy decision point to control and monetize the dynamic subscriber interaction with IP-based services. Vendor-neutral and access-network agnostic, Bridgewater Systems' comprehensive policy management portfolio features network access control products, including authentication, authorization, and accounting (AAA) and dynamic host configuration protocol (DHCP) systems; entitlement control products to manage subscriber access to applications and network resources; and robust subscriber management via a centralized policy and profile repository solution. Bridgewater Systems' proven carrier-class products help Service Providers enrich the subscriber experience and enable extensive revenue capture capabilities and out-of-the-box value that can be deployed in weeks — instead of months.

More than 90 leading Service Providers around the globe, including Verizon Wireless, Sprint, Bell Mobility, and Virgin Mobile USA, trust Bridgewater's technology and business insight to help them deliver world-class services.

Founded in 1997, Bridgewater Systems is a privately held company.

### BRIDGEWATER SYSTEMS

© 1997–2007 Bridgewater Systems Corporation. All rights reserved. Bridgewater, Bridgewater Systems, the Bridgewater Systems logo, Widespan, and One View. Infinite Possibilities are trademarks of Bridgewater Systems Corporation. Other company or product names referenced may be the trademarks or registered trademarks of their respective holders.  
[WWW.BRIDGEWATERSYSTEMS.COM](http://WWW.BRIDGEWATERSYSTEMS.COM)

#### HEADQUARTERS

303 Terry Fox Drive, Suite 500  
Ottawa, Ontario  
Canada K2K 3J1  
Phone: +1 613 591 6655  
Fax: +1 613 591 6656

#### EUROPEAN OFFICE

200 Brook Drive, Suite 102  
Green Park, Reading, Berkshire  
United Kingdom RG2 6UB  
Phone: +44 (0) 118 925 3298  
Fax: +44 (0) 118 925 3299

#### ASIA PACIFIC OFFICE

04–13 Technopreneur Centre  
Block 1003 Bukit Merah Central  
Singapore 159836  
Phone: +65 6276 3447  
Fax: +65 6270 3781

#### U.S. OFFICE

3959 Electric Road, Suite 357  
Roanoke, Virginia  
United States 24018  
Phone: +1 540 772 3103  
Fax: +1 540 725 1067