



PsiNaptic Inc.
216, 200 Rivercrest Drive SE
Calgary, Alberta
Canada T2C 2X5
T: 403 720-2531
F: 403 720-2537

Jini Services versus Bluetooth Profiles

A Bluetooth profile is basically a specification for describing some standard capabilities of a Bluetooth-enabled device. Profiles define what a device can do, and determine which devices can talk to each other and how that communication takes place. Profiles are used to achieve the functionality required to implement a desired application.

Bluetooth consists of numerous protocols encompassing various layers of the OSI protocol reference model, including the physical and transport layers. A Bluetooth profile specifies which Bluetooth protocols are implemented and how they behave. A Bluetooth-enabled device only needs to support the protocols specified by the profiles it implements. For example, the PAN profiles specify a TCP/IP communications scheme for Bluetooth, and a device implementing a PAN profile must contain SDP, L2CAP, and BNEP. A profile may specify certain protocol parameters and may define the format of certain data packets.

One should not compare Jini to Bluetooth as a whole, because they operate at different levels. Bluetooth (with the PAN profiles) offers a transport over which Jini can operate. It seems most appropriate to compare Jini with the Bluetooth profiles themselves as many Bluetooth solutions can be implemented with either a Bluetooth profile or with Jini over a Bluetooth PAN.

A Jini client must have some knowledge about Jini services of interest to it, and the client must know how to utilize these services. This is accomplished using attributes containing Java interfaces. Jini services are specified by Java interfaces. Below are some relevant items comparing Jini Services to Bluetooth Profiles.

1) Ease of Specification

This is the greatest advantage of Jini. We believe that a Java interface is much simpler (and quicker) to specify than a Bluetooth profile. The behavior of the Jini service "driver" on the client does not need to be specified as it can be supplied by downloadable Java bytecode when needed. It is expected that this will greatly reduce time-to-market for new applications.

Developers are free to implement their applications as they see fit, as long as they adhere to the specified interface. Bluetooth profile specifications require much more consideration and will take much longer to develop.

2) Infrastructure

A Jini client must have a JVM capable of dynamic class loading, however, with the advent of CMatos, a JVM is not required on the Jini service provider. Both the Jini client and the Jini service provider must support the Bluetooth PAN profiles. (Currently, the only transport mechanisms specified by Jini is TCP/IP, however, the benefits of Jini could be realized without the use of TCP/IP, by specifying alternate transport mechanisms.) A simple profiles-based solution would likely require less Bluetooth infrastructure, as TCP/IP and BNEP may not be required. For Jini, this penalty seems minor, as there are tiny Bluetooth devices available today that supply the appropriate infrastructure for Jini-based solutions.

3) Network visibility

Jini services are visible and accessible to the larger TCP/IP network. The TCP/IP network may be much larger than a Bluetooth PAN. A profiles-based service will not be visible beyond the Bluetooth piconet.

4) Code Portability

The Jini client "driver" code does not need to be stored on the Jini client. It can be downloaded as needed and then removed when no longer required. This code can be provided by the service provider, a third device, or even reside on the client itself. The profiles-base approach would require that a client store the "driver" code used to implement the various supported profiles.