ORCAjs open real-time communications API

Developer Freedom
Network Independence

Built expressly for web developers, the Orca.js developer framework makes it easy to add Real-Time Communication to web applications—and gives the freedom to choose between multiple service providers.

Simple and Powerful

To quickly grow market share, developers need access to powerful new innovations to ensure their apps perform their best and user experience is optimized. The complexity of the underlying communications signalling infrastructure can be a distraction when adding new features with real-time communications. With the WebRTC standard developers still face the challenge of either building their own real-time signalling infrastructure or buying "WebRTC as a service" from a third party and risking losing control, lock-in or unpredictable quality.

The Orca.js client framework solves this problem with just 20 lines of code that allow developers to simply add real-time communications to applications. With Orca.js the application developer retains complete control over the service behavior and can adapt it to the precise needs of their market. The ORCA API is designed to work with multiple service providers reducing integration cost, and freeing developers to invest in customer experience improvements instead of back-end interfaces.

Flexible

The Orca.js framework allows developers to write an application once and use it, unmodified (or with only minor modifications), with any ORCA-compliant WebRTC service. This has been achieved by bringing together web start-ups, applications developers, major service providers and infrastructure vendors to define an open client-side API for real-time communications. Using a common framework frees application developers from having to worry about the "signalling gap" in the HTML5 WebRTC specification.

The API is extensible and provides tools to enable rapid prototyping and testing of applications. The involvement of major service provider ensures national and international availability with high reliability and break-in/break-out to the PSTN.

Available

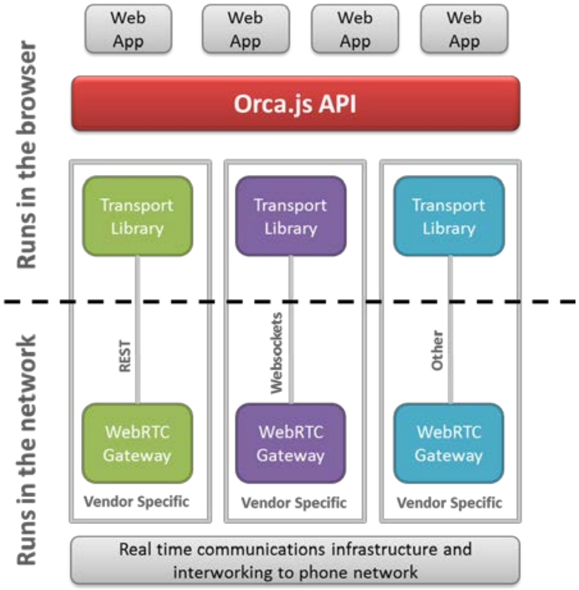
The Orca.js framework is defined in open source Javascript files and example applications that are available today. Development and testing of applications is supported by another open source tool – the "reflector" – which allows Orca.js applications to operate in a private developer environment without the need to connect to a complete signalling backend.

The introduction of commercially hosted ORCA-compliant services is imminent. Developers can also exploit the Orca.js framework with their own infrastructure to offer flexible future deployment options. ORCA-compliant APIs are simple and accessible, with the support of industry leaders to enable WebRTC developers to exploit the full potential of WebRTC with confidence and ease.

For more info and to download Orca.js visit www.orcajs.org.

Technical View for Developers

WebRTC standards in HTML5 are integrating Real-Time Communications (RTC) fully into the web. RTC can now operate with no plug-ins and no installers. This helps users enjoy frictionless access to real-time services, thus opening up new and exciting possibilities. However, from the developer's perspective, the WebRTC standards are not so simple. The lack of a standard signalling protocol leaves developers with a tough choice: either to define their own signalling and build their own infrastructure, or to take a pre-built solution from a third party, which can severely limit flexibility and create a single-supplier dependency.



Orca.js framework

The Orca.js framework is a new approach that works with WebRTC to provide developers with flexibility while retaining the simplicity of application design.

Orca.js API

The heart of the Orca.js framework is the open source JavaScript Orca.js API. Using the API developers can quickly add real-time communication services to JavaScript applications. Unlike some other real-time communications APIs, the Orca.js API provides developers with full control over how communication services are handled. Applications that only need a simple "click to call" service can be written in just a few lines of code, but application developers can also create more sophisticated services by defining their own handling of call events or incorporating other elements like presence and subscription.

The unique strength of the Orca.js API is that it is not tied to a single service provider or network architecture. The API is standard and constant across providers meaning that the same API calls work on different service providers. Even if a particular service provider doesn't support the Orca.js API today it is often practical to provide a mapping to the Orca.js API, meaning the framework is still relevant.

The ability to port applications between different service providers is enabled by the "transport library" part of the Orca.js framework. This library maps the Orca.js API to the specific signalling used by the service provider. Transport libraries may be developed by the service provider or by application designers if their service provider doesn't already provide an ORCA transport library. Only the transport library is tied to the specific service provider. To change service providers, just swap to a new transport library and keep going.

Community

Orca.js has a strong community that is committed to open source and support of developers and service providers using the Orca.js framework. An initiative of ATIS (Alliance for Telecommunications Industry Solutions)—a leading technology and solutions development organization creating innovations to address the industry's most pressing challenges— the ORCA Project brings together service providers, infrastructure providers and developers. For more information on ORCA contact us at info@orcajs.org