

# Scenario: Bluehoo

For more information visit <http://www.azure.com>

**Sentient is a UK based ideas incubator that has developed a social discovery and anonymous messaging application called Bluehoo.**

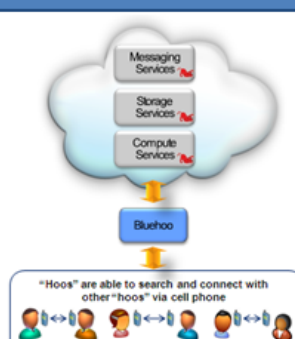
## Situation

Bluehoo has created and deployed a social discovery and messaging application that lets people use their mobile phones to find other users around them with similar interests. To ensure that they have a reliable and scalable foundation for their business, Bluehoo built their application on the Azure Services Platform.

## Problem

To build their application, Bluehoo faced several challenges:

- **Elastic Platform:** Bluehoo could not predict the traffic their new application would drive and needed a platform that could scale dynamically.
- **Shortest Time to Market:** They sought a toolset that would help them take the shortest possible path between concept and product launch with the highest quality and lowest risk.
- **Developer Skills:** Bluehoo wanted to use their existing .NET development skills and not have

Problem	Solution	Benefits
<ul style="list-style-type: none"> <li>•Growth limited by infrastructure</li> <li>•Get to market quickly to establish first mover advantage</li> <li>•No time to learn something new</li> <li>•Want to align to a trusted vendor</li> <li>•Self-funded start-up without significant infrastructure budget</li> <li>•If users don't perceive the system to be fast, they won't use it</li> </ul>	 <p>The diagram illustrates the Bluehoo solution architecture. At the top, a cloud contains three services: Messaging Services, Storage Services, and Compute Services. Below the cloud is a box labeled 'Bluehoo'. At the bottom, a group of mobile phone icons represents users, with a text box stating: '"Hoos" are able to search and connect with other "hoos" via cell phone'.</p>	<ul style="list-style-type: none"> <li>•Business can grow globally and is not limited by infrastructure</li> <li>•Development time is shortened by efficient developers</li> <li>•Developers can leverage existing .NET skills without much ramp-up</li> <li>•Confidence in the platform</li> <li>•No infrastructure capital expense</li> <li>•Asynchronous processing with message queues for speed</li> </ul>

to learn new tools or programming languages.

- **Vendor Trust:** They wanted a vendor and platform that they were familiar with and that they could trust. They were not willing to gamble on a partner whose primary business was not aligned with operating systems and software development.
- **Tight Budget:** As a startup, Bluehoo needed to minimize capital expenditure. Costs were all the more important since they did not want to rely on Venture Capital funds.
- **Performance Speed:** The application and platform needed to be fast to meet the demands of their end users.

## Solution

Bluehoo used the Windows Azure Compute and Storage Services to build a reliable and scalable

service-oriented application. End users (called "Hoos") use Bluehoo to search for other "Hoos" in their vicinity to discover and initiate a conversation. For example, users can arrive at a conference, see a list of people with similar interests pop up on a cell phone, and initiate a dialog around shared topics of interest.

## Create

The Bluehoo application has four major components:

- **Bluehoo Mobile:** Written for Windows Mobile or Java-compatible mobile phones, this application uses Bluetooth technology to scan for other "hoos" in the vicinity. When it finds a "hoo", it communicates this information to the Bluehoo backend using REST API's. BlueHoo, matches profiles and facilitates communication between hoos.

- **Bluehoo Gateway:** This thin web service runs on the Azure Services Platform. Its job is to facilitate efficient communication between the phone and the Bluehoo backend services. When it receives a message from the mobile phone, it puts that message into a queue and returns a response to the phone. It uses asynchronous communication to make response times as fast possible.

The Bluehoo Gateway also queues up outbound messages. When the phone connects, the REST API pulls messages from the queue and routes them to the appropriate receiving phone. To easily deploy as many instances of this Gateway as needed, Bluehoo wrote it in ASP.net using Visual Studio 2008.

- **Bluehoo Services:** These services are “always on” and run on Compute. Compute implements the Bluehoo business logic and communicates with clients using a message queue. Bluehoo adds instances of these services as needed to increase capacity.
- **Bluehoo Website:** The Bluehoo website provides a portal for marketing, software download and support for end users. It is also implemented on the Azure Services Platform in ASP.net.

Bluehoo developers built their application using the Azure Services Platform. They also utilized Silverlight and Virtual Earth. Bluehoo built and debugged most of their application within the familiar Visual Studio 2008 development environment. This helped shorten their development cycle to less than six months.

## Deploy

Bluehoo developers tested & debugged their application in Visual Studio 2008 using the local Azure Services Platform container that emulated the cloud environment. Once the testing and debugging was complete, the application was deployed using Visual Studio 2008. Bluehoo can now maintain their application offering using the Azure Services Platform web portal. There, they can view their services and provision capacity depending on demand. When the team needs to fix a bug or deploy a new feature, they simply make the change on their desktop, test it using the Azure Services Platform container, and deploy the application to the Azure Services Platform.

## Extend

To market their offering, Bluehoo created an interactive marketing website using Microsoft® Virtual Earth™ and Microsoft® Silverlight. These services helped the team provide a rich experience for the end user without having to develop their own media streaming & mapping technology. Bluehoo also chose to use Microsoft’s online tools for email and productivity. This eliminated the need to procure any servers and helped reduce capital expenditure. By not having to worry about operational constraints, Bluehoo could dedicate their attention to addressing business needs and building a rich service. The Azure Services Platform also makes it easy for Bluehoo to provision capacity based on market demands, and pay for only what they consume.

## Benefits

- **Dynamic Scaling:** The fabric controller technology in the Windows Azure platform enabled Bluehoo to scale or upgrade web services seamlessly, as demand rises and falls.
- **Streamlined Development & Maintenance:** Bluehoo developers manage their services in one place, have access to tracing and logging, and can deploy changes quickly. They also do not have to deal with patching the operating system. As a result they spend less time on maintenance and more time writing code.
- **Developer Productivity to Focus on Logic:** Bluehoo developers are more productive because they can build and debug everything – the mobile application, the website, and services – right on their desktop within the familiar Visual Studio 2008 environment.
- **High Availability:** The Azure Services Platform and storage are highly available and redundant. Bluehoo developers do not have to worry about infrastructure outages and could focus their time on building their application.
- **Cost Effective:** The pay-as-you go model minimizes capital expenditures.
- **Superior User Experience:** Azure Services Platform supports a topology which enables fast, asynchronous communication between the mobile application and Bluehoo services, making the application fast and responsive.