

Microsoft® Research

Faculty Summit 2010

Hawaii Project Technical Overview

Brian Zill

Microsoft Research

bzill@microsoft.com

Talk Outline

- Overview of Hawaii Platform
- Illustrative Example
- Details of What We've Been Providing
- Primers on:
 - Windows Mobile Development
 - Hawaii Location API
 - Bing Maps API
 - Windows Azure Development
 - Windows Live ID
 - Hawaii Notification API
- Resources

Overview

The goal of the Hawaii Platform is to ease the development of cloud-enabled mobile applications and enable them to reach higher levels of functionality.

Since these applications can get quite complex in their use of multiple external components, we want to provide various underlying services so the students can concentrate on the aspects of their system that are unique to their application.

Cloud-enabled Mobile Apps

These “cloud-enabled mobile applications” are essentially distributed systems involving a potentially large number of components:

- Mobile Device – UI, sensors, computation, storage.
- Web Site or Service – Web UI, computation, storage.
- Authentication – Returning user identity verification.
- Location – Determining mobile device’s location from environmental clues.
- Mapping – Conversion between location representations (lat/long to map, or street address).
- Notification – Resource-efficient messaging.

Example

A “Find My Phone” application to help people find their lost phones.

Scenario:

- User loses their phone (oops).
- Logs into the “Find My Phone” website.
- Issues request for the phone to report in...
- Views the phone’s location on a map.

What's Involved



Bing Maps



Windows Live ID
Authentication



Hawaii Notification
Service



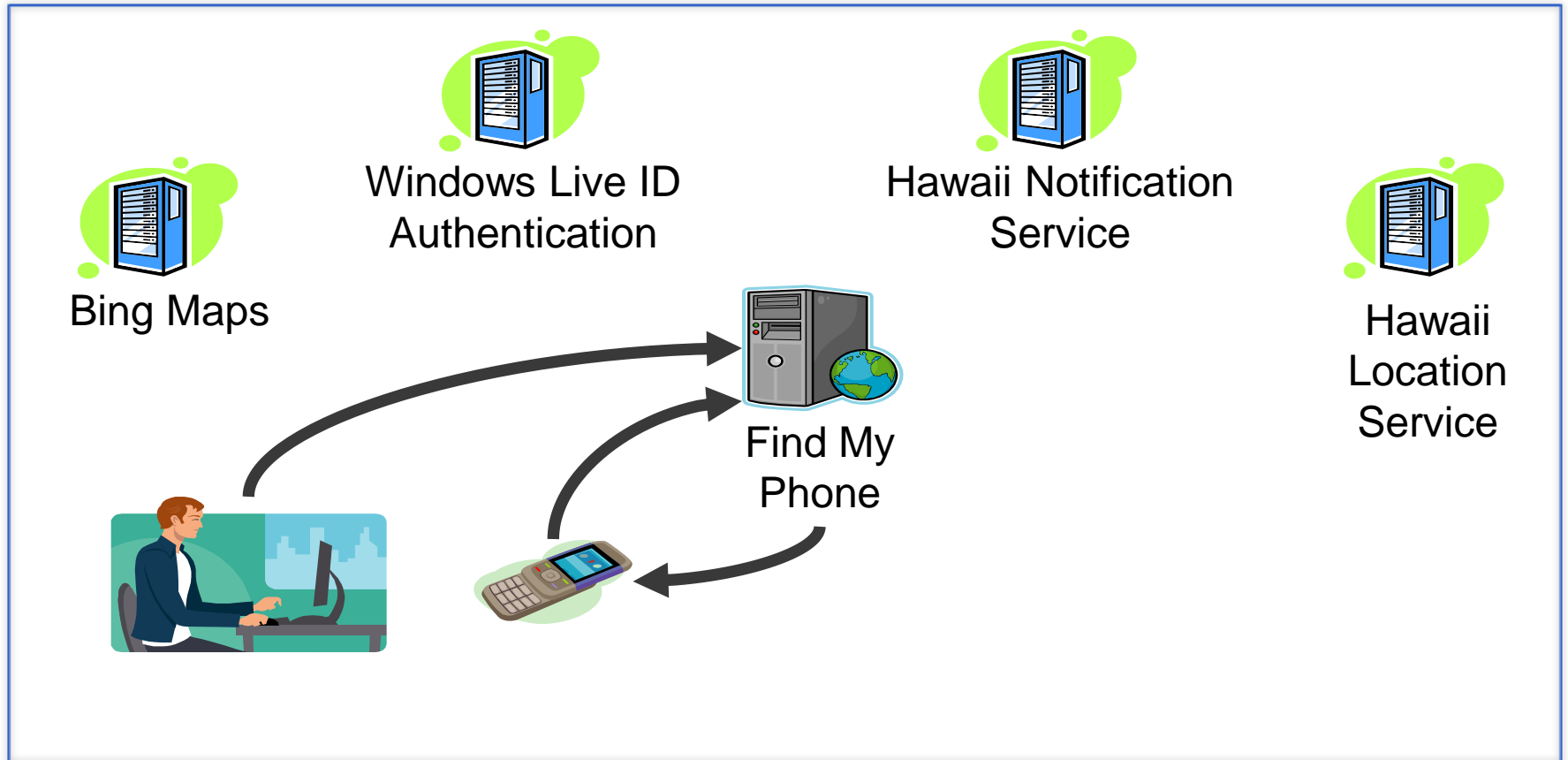
Hawaii
Location
Service



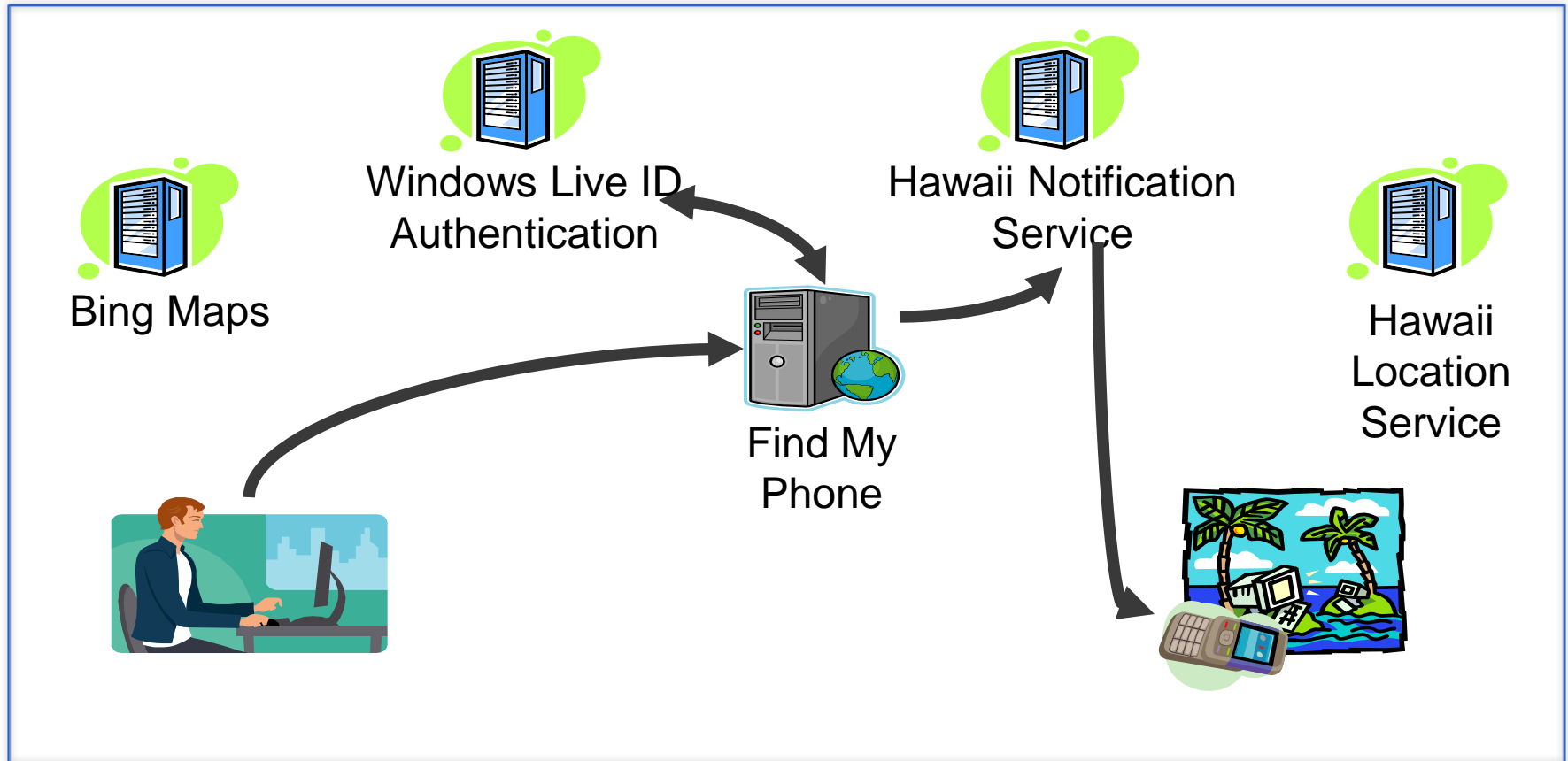
Find My
Phone



Download and Register



Login and Request Update



Phone is Lost



Bing Maps



Windows Live ID
Authentication



Hawaii Notification
Service



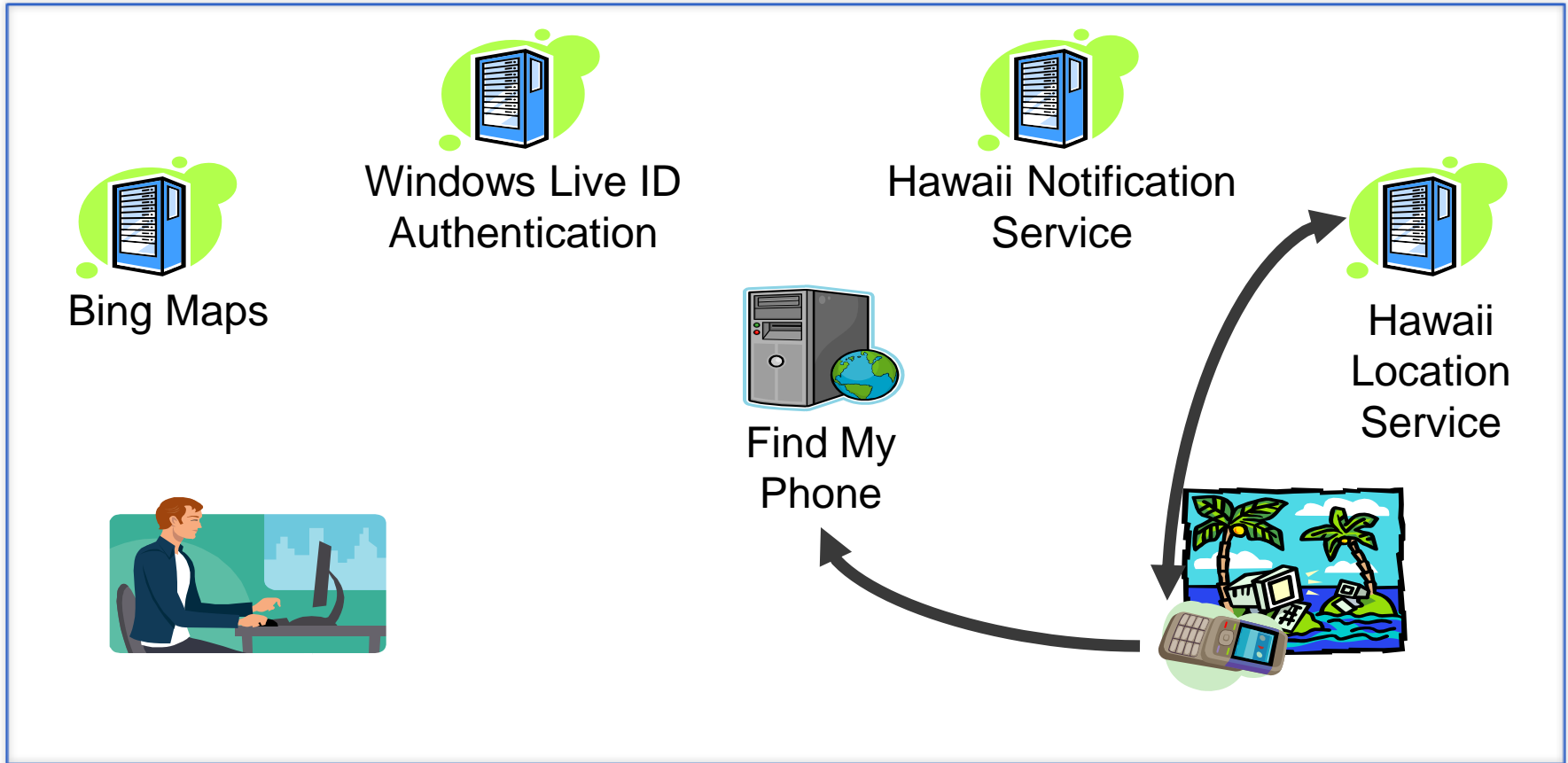
Hawaii
Location
Service



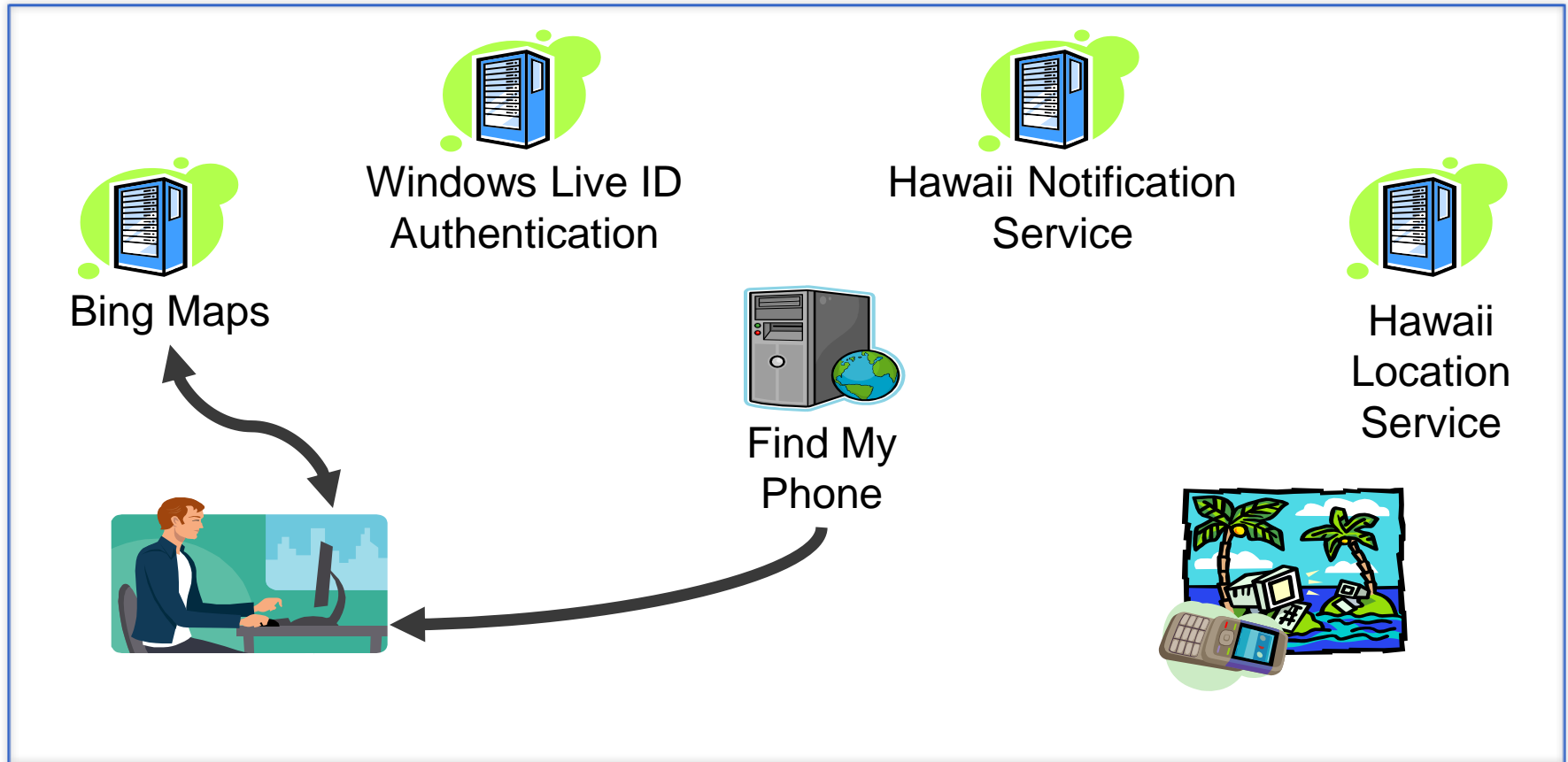
Find My
Phone



Phone Phones Home



User Maps Location



Hawaii Platform Details

We provide phones, tools, SDKs, sample code:

- HTC Pure
 - A touch-screen smartphone running Windows Mobile 6.5.
- Windows Azure
 - Hosted utility for web site/service and database services.
- Windows Live ID
 - Web login service.
- Hawaii Location Service and mobile device API
 - Provides current Latitude/Longitude/Altitude.
- Bing Maps
 - Provides maps and imagery for a particular location.
- Hawaii Notification Service
 - A persistent notification channel.

Mobile Device Details

We have standardized on the “HTC Pure” mobile phone for the Hawaii project.

- Windows Mobile 6.5 professional
- 3.2” touch-screen (800 x 480 resolution)
- 3G radio compatible with AT&T’s network
- Wi-Fi (802.11 b/g)
- Bluetooth 2.0 with EDR
- GPS
- 5MP auto-focus camera
- Also: accelerometer, ambient light sensor, FM radio



Windows Mobile Development

- Development Environment
 - Visual Studio 2008 Professional SP1
 - Windows Mobile 6 Professional SDK
 - .NET Compact Framework 3.5
 - Windows Mobile Device Center 6.1
- Similar to programming for the desktop
 - .NET Compact Framework is subset of full framework.

Hawaii Location API

- We provide a prototype service and API
 - Not yet public.
- Simple API for determining location
 - Provides current Latitude/Longitude/Altitude.
- Handles GPS and radio-location
 - Collects location information from local sensors (GPS readings, WiFi AP bssids, cell phone towers, etc).
 - Back-end service provides algorithms/database for determining location based on sensor information.

Bing Maps API

- Provides information about a location.
 - Street maps, imagery, etc.
- Translates between location representations.
 - E.g. street address to lat/long, place names, etc.
- AJAX API for calling from browser.
 - Interactive SDK: <http://www.microsoft.com/maps/isdk/ajax>
- Other APIs (e.g. Web Services) available.
 - See <http://msdn.microsoft.com/library/dd877180.aspx>
- Demo of simple map display in a browser.

Windows Azure Primer

- We provide the students with Azure accounts.
- Azure is essentially web and database services in the cloud.
- Hosted ASP.NET web site/service:
 - Looks like IIS with web management interface.
- Websites are given URLs of the form `http://<YourNameHere>.cloudapp.net/`

Windows Azure Development

- Development Environment
 - Visual Studio 2008 Professional SP1
 - Windows Azure Tools for Visual Studio 2008
 - Windows Azure SDK
- “Azure Tools for Visual Studio” streamlines the process of building Azure services:
 - Project to hold Azure configuration.
 - Project to hold standard ASP.NET website.
 - Results can be directly uploaded to Azure.

Windows Live ID Primer

- Live ID provides an authentication service.
 - Handles web site “sign in” for you.
 - Provides an unique ID for each signed-in user.
- Web Authentication SDK
 - Source code for WindowsLiveLogin class.
 - Example code for service callback handler.
 - Example code showing use of API to get user id.
- Must register your web site URL with service:
 - Register callback handler page.
 - Get application id (use when calling service).

Hawaii Notification Service

- We provide an experimental notification service and example code for calling it.
- Provides a simple method of signaling mobile.
- Why a Notification Service?
 - Mobile devices rarely have fixed public IP addresses.
 - Power is limited, more efficient to have single system.
- Apps allocate channel(s) from service.
- Clients subscribe to channel(s).
- Channels can be signaled, waking clients.

Developer's VM

- Configuring a dev machine can be time-consuming and problematic...
- We're creating a Virtual Machine containing all the development tools, SDKs and sample code pre-installed.
 - Windows Server 2008
 - Visual Studio
 - All the SDKs, sample code, extra tools, etc.

Hawaii Forum

- Place for participants to contact us or each other to ask questions or share knowledge.
- Discussion Forum on our community server:
 - <http://community.research.microsoft.com>
 - Hawaii Project forum is private.
 - Please create an account for yourself:
 - See “Join” link at top of main page.
 - Email us (hiforum@microsoft.com) your account name and I’ll add you to the forum’s access list.
 - <http://community.research.microsoft.com/forums/149.aspx>

Application Depot

- Place for participants to share their creations with other participants.
- Essentially a catalog of applications with their download URLs.
- Accessible via website or mobile app.
- Mobile app will auto-install the downloaded application.

Hawaii Project Website

- Public website for general info:
 - <http://research.microsoft.com/hawaii>
- Informational website for participants:
 - <http://research.microsoft.com/en-us/um/redmond/projects/hawaii/private/>
 - Has additional information and example code download.
 - Serves as repository of cumulative wisdom.

Microsoft® Research

Faculty Summit 2010

Appendix

Random App Ideas

- Pothole Reporter - App for reporting potholes (w/ pictures) to a website that can display their location on a map.
- Directional Poke - Ping people in a particular physical region.
- Parking Assistant - Find public parking near you, tell others about open parking spaces, find your parked car.
- Taxi Fare Predictor - Crowd-sourced database and logic for estimating taxi fare between two points.
- Commute Logger - Determine which of your alternate routes is best for a particular day of week and time of day.
- Crowd News Reporting - Let a news site alert you to newsworthy events near you so you can cover the event.
- Virtual Graffiti - Post notes that only appear to other people who later visit the same physical location.
- Workout Monitor - Use the accelerometer to determine things about a person's workout.
- Lecture Review - App and website for sharing reviews of a lecture with the other people present.
- Physical Presence Proof - Some means of proving your presence in a physical location, such as responding to some sort of challenge/response.
- Battery Monitor - Crowd-sourced comparison of your battery usage with other peoples to identify when your battery is no longer holding as good a charge.
- Photo Tagger - Geo-tag photos.
- Walking Route Suggestion - Find the optimal way around your College campus.
- Social Heat Map - See if your friends are around, or where they are congregating.

Resources

- **General:**

- Visual Studio 2008 Professional Edition Service Pack 1
 - From MSDN <http://www.msdn.com> or DreamSpark <http://www.dreamspark.com>
- Microsoft Web Platform Installer: <http://www.microsoft.com/web/downloads/platform.aspx>
- .NET Framework 3.5 Service Pack 1 Download:
<http://msdn.microsoft.com/netframework/aa569263.aspx>
- .NET Framework Information: <http://msdn.microsoft.com/netframework>

- **Windows Mobile 6.5 Development:**

- Windows Mobile Developer Center: <http://msdn.microsoft.com/windowsmobile>
- Core Downloads for Windows Mobile Development:
<http://msdn.microsoft.com/windowsmobile/bb264327.aspx>
 - Windows Mobile 6 Professional SDK: <http://go.microsoft.com/fwlink/?LinkId=87437>
 - Windows Mobile Device Center 6.1 (for Vista/7) or Microsoft ActiveSync (for XP).
- .NET Compact Framework 3.5 Redistributable
<http://www.microsoft.com/downloads/details.aspx?familyid=E3821449-3C6B-42F1-9FD9-0041345B3385>

Resources Continued

- **Web Application/Service using Windows Azure:**

- Azure Developer Center: <http://www.microsoft.com/windowsazure/developers>
- Windows Azure Tools for Visual Studio 2008 (includes Windows Azure SDK): <http://go.microsoft.com/fwlink/?LinkId=128752>
- Windows Azure SDK Online documentation: <http://msdn.microsoft.com/library/dd179367.aspx>
- Windows Azure Platform Training Kit: <http://www.microsoft.com/downloads/details.aspx?familyid=772990DA-8926-4DB0-958F-95C1DA572C84>

- **Identification using Windows Live ID:**

- Windows Live ID Web Auth SDK: <http://go.microsoft.com/fwlink/?LinkId=91761>
- Windows Live ID Web Authentication Online Documentation: <http://msdn.microsoft.com/library/bb676633.aspx>

- **Mapping using Bing Maps (formerly known as Virtual Earth):**

- Interactive SDK: <http://www.microsoft.com/maps/isdk/ajax>
- All Bing Maps SDKs: <http://msdn.microsoft.com/library/dd877180.aspx>

Running Your Own Server

- Server 2008 R2
 - Has IIS 7.5 (non R2 has IIS 7.0)
- IIS 7.0/7.5
 - May need to enable using Server Manager
 - Roles -> Roles Summary -> Add Roles
- ASP.NET
 - Need to enable using Server Manager
 - Roles -> Role Services -> Add Role Services
 - Features -> Add Features -> .NET Framework 3.5.1 Features

Web Sites in IIS

- Use IIS Manager to create web sites
 - Connect to server, Select server -> Sites, Add Web Site...
- Important site properties
 - Name (not externally visible)
 - Application Pool (usually one per web site)
 - Content Directory (usually under C:\InetPub)
 - Binding (defines external access point)
 - Host name, IP address(es), Port, Protocol
- Global IIS configuration kept in XML files
 - %SystemRoot%\System32\inetsrv\config*
- Site-specific configuration kept in XML file
 - <Content Directory>\Web.Config

Privacy Concerns

- Users don't like it when programs expose potentially private information to others without telling them.
 - Some countries/jurisdictions have legal requirements.
- **Your applications MUST get the user's consent before calling the location API.**
 - Once the user has consented, your app may remember this fact rather than ask on each run.
 - Exact language required is specified in the SDK.

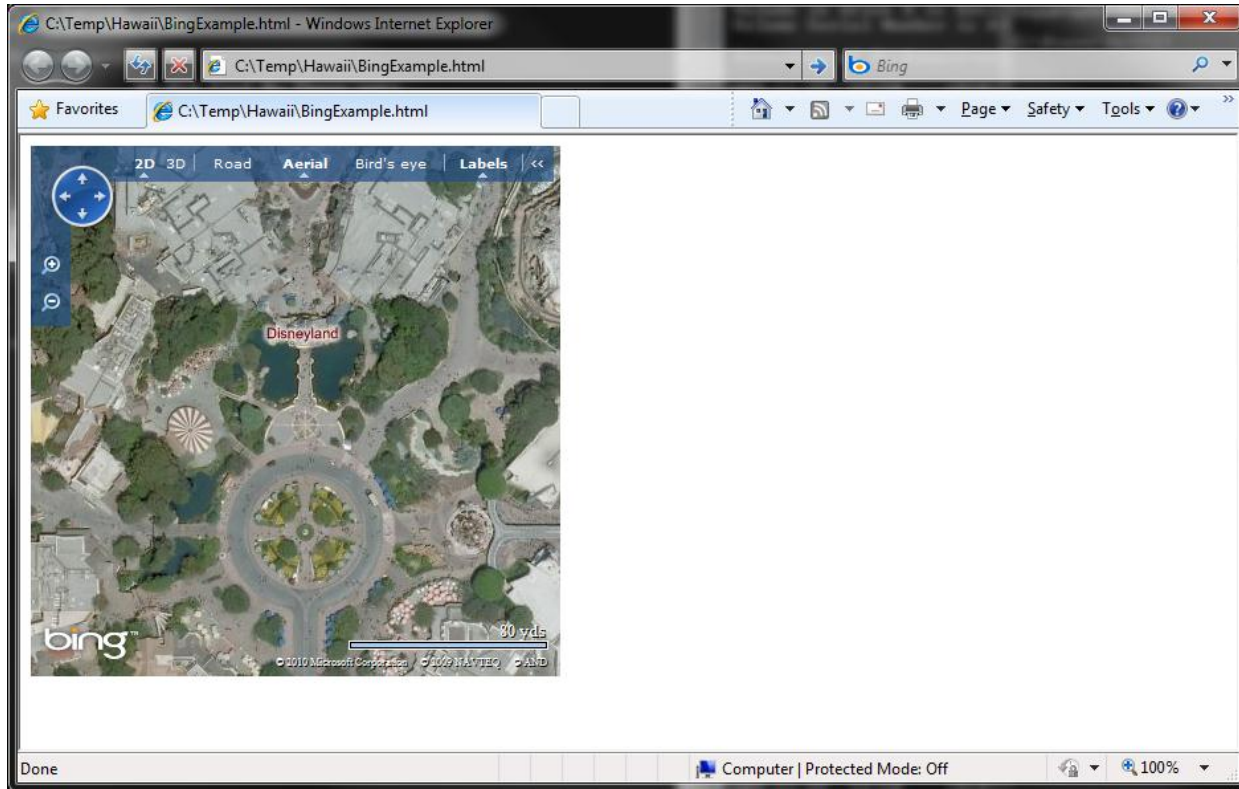
Location API Details

- LiveLocation.CAB
 - Installs on the phone.
 - Extends the OS to include location capability.
- Applications link with LiveLocationLib.dll.
- LiveLocationWrapper provides a .NET API.
- API can be called two ways:
 - One-shot.
 - Provide asynchronous callback updates.
- Demo of simple location app.

Bing Maps Example

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <script type="text/javascript"
      src="http://ecn.dev.virtualearth.net/mapcontrol/mapcontrol.ashx?v=6.2"></script>
    <script type="text/javascript">
      var map = null;
      function GetMap()
      {
        map = new VEMap('myMap');
        map.LoadMap(new VELatLong(33.8125, -117.919), 18, 'h', false);
      }
    </script>
  </head>
  <body onload="GetMap();"
    <div id='myMap' style="position:relative; width:400px; height:400px;"></div>
  </body>
</html>
```

Bing Maps Example



ASP.NET Environment

- Each Web site is an “Application”.
- Applications isolated by .NET runtime:
 - Runs on App Pool thread(s).
 - Created upon first HTTP request.
 - Can timeout if no requests (default 20 mins).
- Most code runs as handlers to HTTP requests:
 - The “Page_Load” event being the most common.

ASP.NET Handlers

- Application/Session event handlers
 - Specified in global.asax file.
 - Application/Session start, end, error, etc.
- Page handlers for HTTP requests
 - Filename.aspx (for layout and optional code).
 - UI objects may be declared in layout and manipulated in code.
 - Filename.aspx.cs (optional place for code).
- Raw HTTP handlers
 - Mapped to arbitrary parts of namespace.
 - E.g. <http://website.com/PurchaseOrder/>*
 - Asynchronous or Synchronous.

Keeping State in ASP.NET

- Application State
 - Defined on Application instance.
 - "Application" dictionary object of name/object pairs.
- Session State
 - In process, state server, SQL server options.
 - "ViewState" dictionary object of name/object pairs.
- Page State
 - Instance of Page object created per request.
 - Only static members of object persist across calls.