

HDC GETTING STARTED PART 6: CLOUD-SIDE APPLICATIONS OVERVIEW — VIDEO, 07:47



Copyright Notice

Copyright © 2022 Wind River Systems, Inc.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means without the prior written permission of Wind River Systems, Inc.

Wind River, Simics, Tornado, and VxWorks are registered trademarks of Wind River Systems, Inc. Helix, Pulsar, Rocket, Titanium Cloud, Titanium Control, Titanium Core, Titanium Edge, Titanium Edge SX, Titanium Server, and the Wind River logo are trademarks of Wind River Systems, Inc. Any third-party trademarks referenced are the property of their respective owners. For further information regarding Wind River trademarks, please see:

www.windriver.com/company/terms/trademark.html

This product may include software licensed to Wind River by third parties. Relevant notices (if any) are provided for your product on the Wind River download and installation portals:

<https://delivers.windriver.com/>

<https://windshare.usa.windriver.com/>

Wind River may refer to third-party documentation by listing publications or providing links to third-party websites for informational purposes. Wind River accepts no responsibility for the information provided in such third-party documentation.

Corporate Headquarters

Wind River
500 Wind River Way
Alameda, CA 94501-1153
U.S.A.
Toll free (U.S.A.): +1-800-545-WIND
Telephone: +1-510-748-4100
Facsimile: +1-510-749-2010

For additional contact information, see the Wind River website:

www.windriver.com

For information on how to contact Customer Support, see:

www.windriver.com/support

HDC Getting Started Part 6: Cloud-Side Applications Overview — Video, 07:47

29 January 2019

1. HDC GETTING STARTED PART 6: CLOUD-SIDE APPLICATIONS OVERVIEW — VIDEO, 07:47

Published on 6 October 2015

Transcript

Time (mm:ss)	Narration
From: 00:11	Alright, in this section we're going to go back and review this slide that we looked at earlier. You'll notice we've been spending most of our time down at the bottom, in that blue box at the bottom that says "Embedded Software." This is where we've been defining applications that run on the target device and we are able to then invoke them through the cloud console application, but, we've been spending our emphasis on that lower side of the development. Now what we're going to do is we're going to take a look up at the top, the cloud side application development, and we're going to talk a little bit more about what's necessary there and what capabilities are available to us there. And, as an example of what this might look like, you can see that the application development, provided by Wind River,
From: 01:00	as APIs at the top, and there are APIs at the bottom, and the applications developed at the bottom is what we've been doing. However, now we're going to look at developing applications at the top. And this is where the customer in the middle column of this diagram would develop an application that they might present to an end user. The Web console that we've been using is really kind of more for the developer or operations kind of folks, but if you want to develop on the custom application, perhaps an application that would run in your phone, or on a tablet, or in, in a normal browser type application, this is how you would do this is through these APIs at the top side of the platform. Here's kind of an illustration of what you can see over at the top left, the Web console that we've been working in, and this is where you go in and you define your various capabilities,
From: 02:00	define your model types, and to define the actions and rules. You can set that up, but then what you want to develop a Web application or something, and this is where in the middle column, you might develop an application at the bottom that displays something on a target device, but we're going to focus now on looking at the top where we're developing an application to run in the cloud, in a Web browser, and we're going to implement something in JavaScript, and we're going to see how that works. And how does this actually work is really the cloud platform, the Helix Device Cloud, provides a RESTful API so that you can implement your applications by making REST or SOAP calls into the platform, and so you would develop an application to run standalone on another server, or on your desktop from a browser, whatever, and those, those would simply go through our Mashery API Management Tool. The Mashery API Management Tool,
From: 03:03	you know, gives you an additional set of capabilities for managing the calls through the applications into the cloud platform so that you could manage throttling of the number of calls that are being used, or logging metered billing kinds of applications, and the capabilities over in the platform we've talked about a little bit already, but we can also extend those with custom objects and to define additional capabilities in the platform. And so the API itself, you know, is a programmatic API that runs over an HTTP query. So you're doing simply gets, and puts, and posts as you would for any Web-type application development, and so we're going to, we're going to take a look now at what that gives us. Here's kind of an example, illustration, of the application that we'll walk through here. You can see from a browser application,
From: 04:02	we're going to go to make calls into a Node.js server in this example that will have some JavaScript application running there, and it will respond back to the browser with HTML and

	JavaScript that needs to be running in the browser, and then it will send those requests then back to the Node.js server from the browser, and that will then be transformed into these API calls that will go through Mashery and onto the Helix Device Cloud platform. And those will be, you know, service requests for finding assets, or finding alarms, and whatever other capabilities that we may want to query. The returns are obviously either JSON or XML can come back, and the browser would then render that in whatever format was determined necessary. Over on the far right you can see I've listed some custom objects. The cloud platform provides the ability to define additional extended capabilities or
From: 05:04	custom objects, and those would be written in either Groovy Scripts, Java code, and those can be loaded into the platform, and so you can then invoke those through the RESTful APIs as well. Also, we're going to take a look at using these RESTful APIs is the technique that you would use to integrate with other enterprise applications. So if you want to make connections, send data on, this is, this is the same mechanism that you would use here as well. The techniques here, again, you're going to be able to define some of these custom objects, and then in those custom objects you can take advantage of all these capabilities of, you know, defining alarms or monitoring alarms, and setting up timers, and all kinds of rules that you can define, and so these
From: 06:00	are the hooks that give us the ability to connect with third-party applications. The Groovy scripting is the primary tool that's used for defining these custom object applications that will run up in the cloud platform. Groovy is a scripting language that is an extension of Java, but they provide the ability for you to develop these custom objects in this Groovy scripting language, and then simply load them into the cloud platform, and utilize these capabilities. The Scripto Webservices is an API that you can then call those from your application to use these extended objects in your own applications. Here is kind of an illustration of a typical scenario where the customer wants to create a Web browser application that would manage the device through one interface, going through to the device management, but maybe you have
From: 07:00	private data centers where you wanted to store other data, and you could then instruct, through the cloud platform, to send messages down to the agent. The agent could then invoke other custom written code that would then probe devices, collect data from sensors, whatever is necessary, and return that data either up into the cloud platform, and even forward it back onto the browser so you can display it in the browser. But you could also send that data off to private data centers and manage it through that interface as well.

Contact: nlyons

Content ID: 045833

template('WindRiver/function/JS/wrConditionalContent');