Diary of an Embedded Developer

Aaron Foellmi
Vice President, Sales and Marketing
The RTC Group
July 07, 2014

Chapter 1 (Starting Out)

I just got out of a strategic meeting. Looks like our company is going in a new direction. We've been working on a new OEM medical device, but management wants us to expand our horizons by designing a server appliance with connectivity to control the entire solution ecosystem. I've set up enough servers in my day, and certainly our expertise with edge embedded devices is considerable—I just worry about pulling it all together seamlessly.

One of my colleagues raised an interesting question—***How do we manage all this data?*** I think we have our work cut out for us to make sure we bring to market a solution with strong:

- Edge device functionality
- Connectivity between edge, gateway and server appliances
- Data management, security and administration
- Analytics and reporting mechanisms to optimize system performance

With the OEM device already well underway, we've been considering various server solutions. Our concerns are pretty straightforward. Implement an embedded server appliance for our Cloud-enabled application with special attention paid to concerns over solution longevity, support and performance.

I've been through this before. The design specification seems pretty straightforward, but we never appear able to deliver the solution ***on time and in budget***. Our device is going to require access to, and analysis of, large amounts of data from multiple edge devices distributed around the world. Balancing security and performance is going to be a challenge. I like a challenge.

Contents

Chapter 2
(Open Source) 2

Chapter 3
(Cost of Ownership) 2

Chapter 4
(Performance and Management) 2

Chapter 5
(Microsoft Product Availability
and Arrow Support) 3

Chapter 6
(Right Channeling Windows
Embedded Server) 3

Chapter 7
(Success) 4

Chapter 2 (Open Source)

What a disaster. We have been evaluating a “boxed” open source solution, hoping to save a little money on the front end. When has “free” ever meant free? There’s growing frustration with our development team that we’re just spinning our wheels.

With a medical device, data security is a key concern. We need to make sure that our server configuration provides centralized access to audit policies and built-in security. With edge devices connected to each other, we have to guarantee the system isn’t compromised.

Setup has proven more difficult than we originally anticipated. We are trying to balance server performance by optimizing the software, and we just can’t seem to get the right combination to unlock this thing. We’re spending an awful lot of time writing custom code. And even though my engineers enjoy the exercise of custom programming, we’re falling behind fast. I don’t even want to think about having to document and support all the proprietary stuff we’re building here. Beyond setup, getting this thing Cloud-ready is an issue.

The servers aren’t ready to go live, and we still haven’t even started thinking about data management and analytics. Yet another set of open source software solutions to build. The required databases will double the documentation and support issues we’re already facing. It’s turning out to be more of a challenge than we expected.

We have excellent software and integration talent within our organization, but what happens as the system evolves, scales and upgrades over time? My colleague joked that at least we’re creating job security with all the support requirements of this application. I don’t share his optimism; if we can’t get this to market faster, we are going to miss our window of opportunity. This is a great application that is going to help a lot of people—but we have to make sure it’s secure and safe. I am really worried that relying on this open source software leaves us standing all alone when bugs and challenges surface.

Chapter 3 (Cost of Ownership)

We have to figure out a way to get this device to market faster. Getting an open source embedded server set up, configured, secure and Cloud-connected is putting us way behind schedule. It is time to look for other options; options that get us to market faster, create stronger technology foundations and don’t require me to spend valuable resources chasing gremlins.

We may have taken the long way around, but I think we’ve come to the right decision to evaluate Windows Embedded Server 2012 R2 and SQL Server 2014. We’re going to kill several birds with one stone. Although the initial purchase is a factor, when we compared the features built into Windows Embedded Server against the time to patchwork similar features together with alternate solutions—Windows Embedded Server offered clear advantages including:

- Ubiquitous Connectivity—We’re adopting a platform that is well known to my development staff (reducing additional training), and that connects edge devices, gateway devices and our new server appliance naturally. This is going to save us from wasting huge amounts of time with custom code, setup and interconnectivity problems.
- SQL Server solves our data management issues and completes the solution. It is Microsoft native software with strong security, analytics and reporting features built in.
- We get to rely on the history and strong reputation of Microsoft software, and can leverage their existing documentation and support to get this solution up and running. And, more importantly, I can count on them to provide updates and bug fixes as they’re discovered. I can sleep better.

We didn’t start off calculating the overall cost of ownership once the system was deployed, but it quickly became apparent that support was going to be a watershed issue for our client. Trying to support open source software for the next decade was going to create too much overhead to keep our solution competitive.

And honestly, it’s good to have partners like Microsoft and Arrow, our distribution partners with a stake in our application’s success.

Chapter 4 (Performance and Management)

Performance is always a concern. Windows Embedded Server provides a platform for easy deployment. So even though we’re only working toward a limited rollout of the application, our goal is to make this a globally adoptable system. We’re eventually going to be dealing with highly distributed edge devices, managing large amounts of data. Although our initial rollout doesn’t require optimized performance, it’s nice to know that as we scale up in the future, we have the ability to rely on enterprise-proven technology like Windows Embedded Server and SQL Server to provide a quick launch today and performance optimization for the future.

Ensuring that our server performance is nimble enough to create a seamless interface for our users, and that management of the system doesn’t cost an arm and a leg, is going to be a feature that our clients appreciate.

We’ve found several key features of Windows Embedded Server that meet specific technical requirements for our application while providing the performance we require.

- Windows PowerShell has helped us reduce errors and improve the flexibility of the overall server appliance.
- Remote server administration and automation tools built in to help manage and automate routine tasks and functionality.
- Data Deduplication allows our data to grow without wasting resources. We’ve been able to maximize our server performance by reducing the amount of duplicate data being stored. Optimizing data storage will allow us to save on proposed hardware requirements along with anticipated storage upgrades as our application moves through its lifecycle.
- With Storage Virtualization Options, we’ve been able to consolidate our disk storage into storage spaces and storage pools to ensure server availability.
- Network adapter teams enable multiple network adapters to be placed into a team interface for bandwidth aggregation and traffic failover.
- CPU and memory throttles balance server appliance workload, which creates a really scalable solution as we expand the penetration of our application within the market.
- Windows Embedded Server and SQL Server create a powerful package to manage data from powerful connected edge devices.
- Hyper-V will be used mange virtual machines.

SQL Server 2012 for Embedded Systems also offers key features worth mentioning:

- Advanced security features to ensure that system data is always protected
- Business analytics and reporting to allow users and the system to gain intelligence from the system data via common querying methods
- Data Quality Services and Master Data Services provide a management framework for IT to centralize structures, object mapping, reference data and metadata
- Ability to exploit Microsoft and other software resources won’t build us into a corner

Chapter 5 (Microsoft Product Availability and Arrow Support)

Beyond the obvious ubiquitous nature of developing on and targeting to similar platforms, the Windows Embedded advantage extends through the whole technology stack. In talking with our Arrow and Microsoft partners, I can see that their strategy moving forward in the embedded space closely matches our own. This includes developing fully integrated systems from small edge devices, to managing gateway devices, to data hubs, all connected via the Cloud . This new intelligent systems concept has so many possibilities, yet it depends on REALLY complex interdependencies. It’s an exciting time in the embedded community.

Choosing Windows Embedded ensures that we have product support for 10 years, which is more than any other server software product that we evaluated. Given the interdependency of these systems, that’s a big insurance policy for our developer team. That along with product availability for 15 years gives me a strong sense that we aren’t engineering ourselves into a corner. Choosing Windows Embedded has exposed our team to technology building blocks that are robust, interconnected and highly stable. There’s a certain confidence that comes from having a couple of heavyweights in your corner.

Arrow has also proven to be more than just a product distribution partner. They’re helping my entire team work through important technology decisions—decisions that will put our solution in the strongest position to deploy on time and on budget not only now, but also in the future. They’ve been able to help us specify hardware and software; work through integration concerns on both the front and back ends, which has allowed our team to move quickly through parts of our design cycle.

Their knowledge of the whole ecosystem has been a huge asset to our team. I’m especially appreciative of their global footprint. They were able to work with our teams in Europe and China to provide comprehensive support, so that we didn’t get hung up on any of the international specific concerns.

Chapter 6 (Right Channeling Windows Embedded Server)

Since Windows Server and Windows Embedded Server are technological twins, we were able to adopt strong and familiar technology AND save significant costs using special licensing designed for our embedded application.

Even though we knew that Windows Embedded was the solution path we were going to take, it was a pleasant surprise to know that we could have Arrow OEM Computing Solutions (OCS) take care of software installation and system configuration on their end as we moved through the procurement process.

Arrow worked with us through their dedicated Microsoft business managers as we were specifying the hardware to build out our server appliance with licensing optimized for embedded solutions. The savings on Windows Embedded versus Windows enterprise grade software is something my managers are going to be thrilled to see, and it didn't require me to make any technology concessions. Just another validation of our overall design strategy!

Chapter 7 (Success)

On Schedule. Two words every manager dreams of hearing. We're on schedule and our server implementation was a success. Arrow's assistance, essentially acting as an end-to-end general contractor for our solution, and Microsoft providing a strong software platform, has made the goal of creating truly intelligent systems far more easy to accomplish.

Lessons Learned:

1. Working with what you already know and use every day makes the job a whole lot easier.
2. Don't forget the database if you need to manage data.
3. Having strong partners jumpstart development and reduce overall cost of ownership for our embedded applications gives us development dominance.
4. The Microsoft Windows Embedded portfolio provides a substantial technology advantage with reliable and time-tested software platforms that we can utilize from the smallest edge device to robust Cloud services. Microsoft is truly an end-to-end solution.
5. Embedded licensing options provide beneficial alternatives to standard licensing, which will reduce solution costs.
6. Arrow fills in the gaps wherever we need them. Being able to rely on their expertise and knowledge base, global footprint and diverse access to tools and products will give us the ability to capitalize on alternatives we may have never considered before.

Footnote—This is a fictional account that exemplifies the features and benefits of using Windows Embedded Server in your next embedded application. Arrow and Microsoft are committed to partnering with you to create the most amazing and intelligent solutions—and Windows Embedded Server 2012 R2 and SQL Server Embedded 2014 provide the data backbone that creates a competitive advantage for OEMs developing devices on the edge with performance-oriented data needs. Arrow and Microsoft can help any OEM become a truly end-to-end solution provider.

Go to http://ocs.arrow.com/msembedded/ for additional resources, or contact your existing Arrow Account Manager to discuss how we can help you supercharge your intelligent solutions.

Use our Get Started link to find out how we can help you beat deadlines and meet management expectations.

ARROW
Five Years Out

Arrow Electronics, Inc.
Marketing Group Goes Here

7459 South Lima Street
Englewood, CO 80112, USA