

From: Yuval Neeman
Sent: Thursday, October 29, 1998 9:24 PM
To: Brad Lovering; John Shewchuk
Subject: FW: Last ditch try

-----Original Message-----

From: Michael Toutonghi
Sent: Thursday, October 29, 1998 3:39 PM
To: Web Win32 Client Server Team
Subject: RE: Last ditch try

I agree with some of what Adam says, but not the conclusion as it relates to our app model. Maybe to the next NT, but not our app model. We are and should not stop building and improving our clients. I recently found out that both Oracle and Sybase have the tiny footprint database that I have been saying we need. Oracle just released theirs, Sybase is in beta. Funny thing is, they expose Java APIs, etc, etc. If we don't build the client side code platform, someone else will get it right. When the killer apps that take advantage of it start coming out (offline, finance, etc.) and Java becomes the accepted client, we will not be able to go back to HTML. If we drive the innovation in this space (which would simply take cooperation), we can make sure that HTML is the way people continue to get reach, and ironically, we can provide more reach for developers than a Java focused client. I don't believe our competition is as positioned as we are to put everything together, but time is running out. Should the next NT be focused on the server and reliability? Absolutely! Should our app model be totally server-centric? No.

I have a model that I believe gives us absolute Windows leverage, huge customer benefit, and more reach than our competitors will have in the very near future. I believe it is the application model we should pursue. It is not in conflict with Adam's suggestions below when it comes to improving the server, but it does leverage clients. I'd be interested in constructive feedback.

Thanks,
Mike



Opportunity.ppt

-----Original Message-----

From: Mohsen Al-Ghosein
Sent: Thursday, October 29, 1998 10:19 AM
To: Adam Bosworth; David Vaskevitch; J Allard; Bharat Shah (NT)
Cc: Michael Toutonghi; Eric Rudder; Mark Lucovsky; Yuval Neeman
Subject: RE: Last ditch try

Needless to say, I agree 100%.

I don't believe that we can have operating systems with multiple centers of gravity. We already do. It's a mess, and it should be. We're trying to do too much with it.

Let's pick one center of gravity and execute on it in the year 2000. I think that it should be the application server. I think that the client is to be assumed to be browsers for the 90% case, and smart Windows clients opportunistically (e.g. coordinating schedules as part of an application with Outlook and a business' web site).

-----Original Message-----

From: Adam Bosworth
Sent: Wednesday, October 28, 1998 11:01 PM
To: David Vaskevitch; J Allard; Bharat Shah (NT)
Cc: Michael Toutonghi; Mohsen Al-Ghosein; Eric Rudder; Mark Lucovsky; Yuval Neeman
Subject: Last ditch try

I sat there yesterday listening with my heart sinking. Why? Well, we have one set of problems which are well understood and for which we theoretically have a solution. These are the problems of building a robust applications/web server. We more or less know what we need to do here even if we assume a DHTML client:

Make it easy to deploy code. Make it easy to make the code safe. Make it easy to write high-performance scalable code in Java or VB directly against the HTTP on the other side with a programming model understandable by VB style programmers. Make it easy talk to mail, to distributed lookup services, to chat services, to data, and to legacy applications

MS/CR 0031955
CONFIDENTIAL

and above all make it easy for unix systems to interoperate with all this. Make it robust and remotely administratable. Rework the model for searching and finding data on the web to support cataloging and schema information so that we can search intelligently for articles on Africa. Fix it to allow 64 bit so we can actually use memory in large quantities efficiently. DavidV and I completely agree about the huge potential ON THE SERVER for in memory databases although I'm worried about scaling. Fix proxying to have quality of timeliness so we can proxy all our data all over the place reliably. Now I know I'm being facile here and just agreeing about what it means to make it easy to talk to data to pick one I'm examining isn't "that" easy. COM+ by itself may not solve all these problems. But it isn't rocket-science either. We can do this. It is hard. It is dirty. It is a lot of work. We need to keep way more serious n-tier customers in the loop from the beginning much as Sql Server has kept some key customers in the loop from the beginning. Plenty or willing like Baan and Merrill Lynch and probably 25 I don't know. But this alone will take a LOT of work. We probably couldn't get all this done with the active support of the tools team until sometime in 2000. It is worth doing though. The web is about two fundamental new kinds of applications, applications to act as intermediaries between customers and data (amazon.com, schwab, online radios, medical research systems, ...) and applications to foster collaboration (talking to people, shared whiteboards, discussions, calenders, clubs). The scope of these applications is such that I confidently predict that for every ISV authoring an application to "author documents", 99.5 will be authoring these kinds of applications.

The problem is that we've decided that we must simultaneously make a "smart client" a lynchpin of our strategy and thus the operating system that ships everything above must also help NT replace Windows 98, fix the stability problems of Windows 98, start to solve the obvious problem that Windows 98 is such a bad net application, and must have 3-tier models in which lots of smart code runs on the client and thus we need all sorts of synchronization mechanisms between client and server for moving data back and forth. This is also a huge effort. BUT it isn't as critical and it may be on the wrong platform. Office users aren't running away from Windows. Sure we have risk, but it isn't as immediate. Games players are still using every API we give them. And maybe, just maybe, Win CE is really the right platform to bet on for the client at least for the revolution.

My point isn't this. It is that we are putting too many eggs in one basket and run the huge risk of doing everything poorly and/or slowly and nothing well. I think senior management MUST make the hard choices here and decide that 2000 is the year when we try to OWN the internet applications server, period. Let's focus on making a version of windows 98 that is as robust and reliable and easy to use as some discrete team can manage and put our wood behind this effort knowing that this is where 99% of the applications/energy/money will be.

Last time I'll evangelize like this (I promise)
Adam Bosworth

Windows Everywhere, Again!

Generating Windows Excitement

MICROSOFT CONFIDENTIAL INFORMATION

Winning the New Platform War

- **Apps are now distributed, we must recast our platform**
 - ◆ We must improve our servers, but our platform includes clients
 - ◆ We are replacing client code in our distributed application model with protocols that require little IP in client, this is bad
 - ◆ We mistakenly think our APIs on client tier preclude reach
 - ◆ Web applications do not leverage client code yet, but they will
- **WORA really just means "Java Everywhere"**
 - ◆ Java is easy, robust, and improving on the server (EJB, IBM SF)
 - ◆ The client threat will resurface with a vengeance
- **EJB + Java Clients == consistent, distributed programming model**
 - ◆ Weak integration of important technologies will improve
 - Forms/HTML
 - Client side, synchronized data (new from Oracle Lite, Sybase)
 - Device support
 - Offline

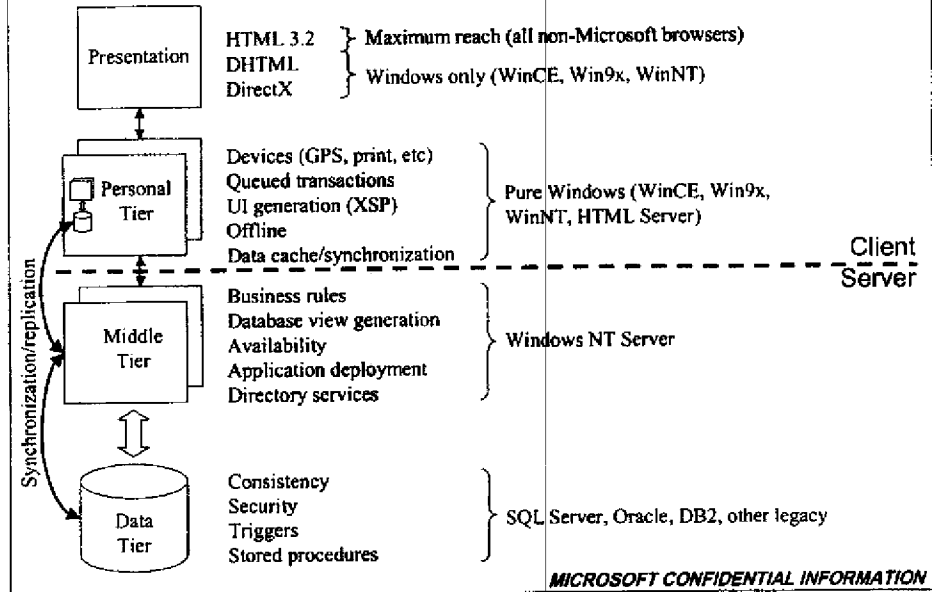
MICROSOFT CONFIDENTIAL INFORMATION

We Must Lead or Follow

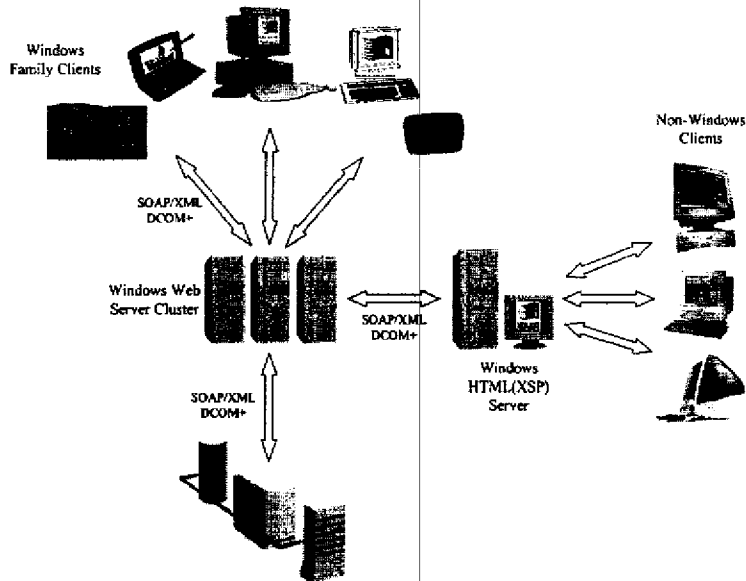
- **Scalability and reliability, yes, but also a model that advantages Microsoft**
 - ◆ **Leverages, requires Microsoft servers**
 - ◆ **Highly leverages, depends on Win9x, WinNT, WinCE clients and Microsoft IP**
 - ◆ **Makes enterprise, web development easier**
 - ◆ **Doesn't force reach/rich binary decision**
- **We should ship by early 2000**
 - ◆ **Understand the core application model**
 - ◆ **Work in a straight line**
 - ◆ **That may not be too late, but it's close**

MICROSOFT CONFIDENTIAL INFORMATION

Windows 4-Tier Architecture



Windows 4-Tier Platform



MICROSOFT CONFIDENTIAL INFORMATION

How We Leverage Windows

- **Personal tier runs application code and is Windows only**
 - ◆ **Data view caching**
 - Synchronized database views
 - Consistent online/offline access model (persistence)
 - Queued batch updating
 - ◆ **Device access (printing, scanning, GPS, vision, mic, etc.)**
 - ◆ **Offline applications, not just UI**
 - Personal Finance (checkbook, eCommerce + commerce)
 - Sales force automation
 - Hotmail, address book, sidewalk.com, shopping, package delivery, real estate, automaps, auto monitor, etc.
- **NT HTML server proxies personal tier through HTML for reach**
 - ◆ Applications designed for reach author once for rich client, leverage and depend on Windows, limit presentation to HTML 3.2
 - ◆ No offline / no code on client (except script)
 - ◆ Screens that use DHTML/DirectX UI cannot be proxied
- **NT Server provides middle-tier <-> personal tier services**

MICROSOFT CONFIDENTIAL INFORMATION

We Need Specific Technology

- **Consistent HTML UI model between servers and clients**
 - ◆ **Flexible persistence and generation model**
 - **HTML compatible form model that fires events out to app, not just inside page**
 - **Persistent UI format that lacks HTML quirks, localizable, etc.**
 - **Request/response event model between code/visible forms**
 - **Screens that target reach can sink get, post, nav events**
 - **Screens that run only on Windows clients can go wild**
 - ◆ **Lightweight HTML depersistence engine (XSP) must scale to WinCE**
- **Isolated code deployment (can be limited to COM+ at first)**
- **Code access security**
 - ◆ **Required for successful client code deployment**
 - ◆ **Useful for server applications like supply chain management**
- **Database (w/view synchronization/update) on client (Win9x, NTWS, WinCE)**
 - ◆ **Oracle has this, Sybase is in beta**

MICROSOFT CONFIDENTIAL INFORMATION

Technology Required (cont'd)

- **Object pass by-value (w/class location info)**
 - ◆ Lightweight remoting infrastructure (must run on WinCE)
 - ◆ Queued calls
- **Compatibility across windows clients**
 - ◆ CPU portability for client code
 - ◆ Common (subset/superset OK) storage / synchronization services
 - ◆ Common (subset/superset OK) display / print services
- **Tools that target the application model**
 - ◆ HTML+ form design
 - ◆ Personal tier/offline tool palette & coding
 - ◆ Server object tool palette & coding

MICROSOFT CONFIDENTIAL INFORMATION

It's Closer Than You Think

- **We could ship this by H1 2000**
 - ◆ **We must clearly articulate the vision**
 - ◆ **We must have a coordinated call to arms**
 - ◆ **We must get started immediately**
 - ◆ **OS implications**
 - **We should rely on Win98 & WinCE for consumers until 2001**
 - **NT 5.1 should focus on reliability, IIS, and COM+**
- **With clear, specific, and consistent directive by our executive staff, we can do this before our competition**

Lets make Windows an exciting platform again!

MICROSOFT CONFIDENTIAL INFORMATION