Solaris in a small business: a developer's perspective

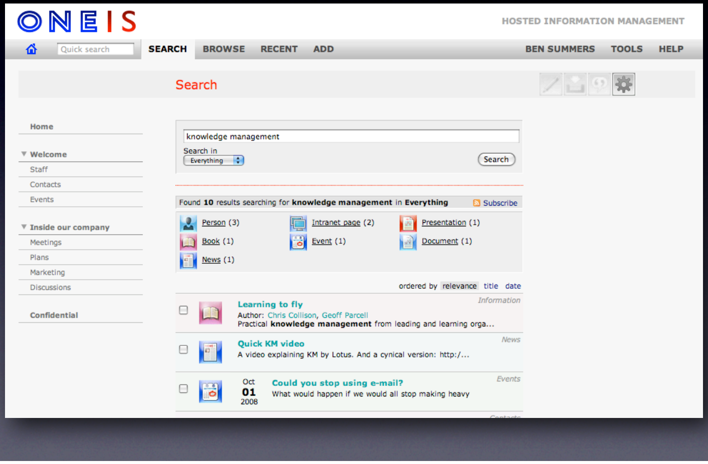
Ben Summers
ben.summers@oneis.co.uk
www.oneis.co.uk/blog

ONEIS™ HOSTED INFORMATION MANAGEMENT

Hello, I'm Ben, and I'm a developer.

As LOSUG is primarily system administrators who work in large organisations, I wanted to share with you my experiences of using Solaris as a developer in a small company.

OneIS (the web app)



knowledge management

Search in: Everything

Found **10** results searching for **knowledge management** in **Everything** Subscribe

Person (3) Intranet page (2) Presentation (1)
Book (1) Event (1) Document (1)
News (1)

ordered by relevance title date

□ **Learning to fly** *Information*
Author: Chris Collison, Geoff Parcell
Practical **knowledge management** from leading and learning orga...

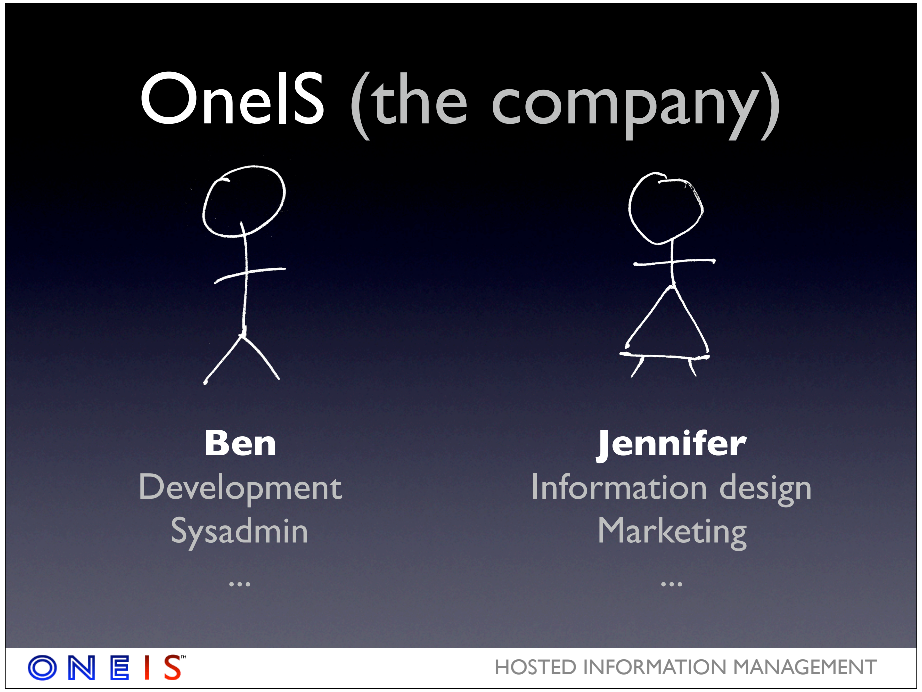
□ **Quick KM video** *News*
A video explaining KM by Lotus. And a cynical version: http:/...

□ Oct 01 2008 **Could you stop using e-mail?** *Events*
What would happen if we would all stop making heavy

ONEIS™ HOSTED INFORMATION MANAGEMENT

At OneIS, we’re developing a web application which helps smaller organisations manage their information.

As well as developing it, we also host the service without outsourcing anything to an ISP. So there’s a lot to do for a small company.



OneIS (the company)

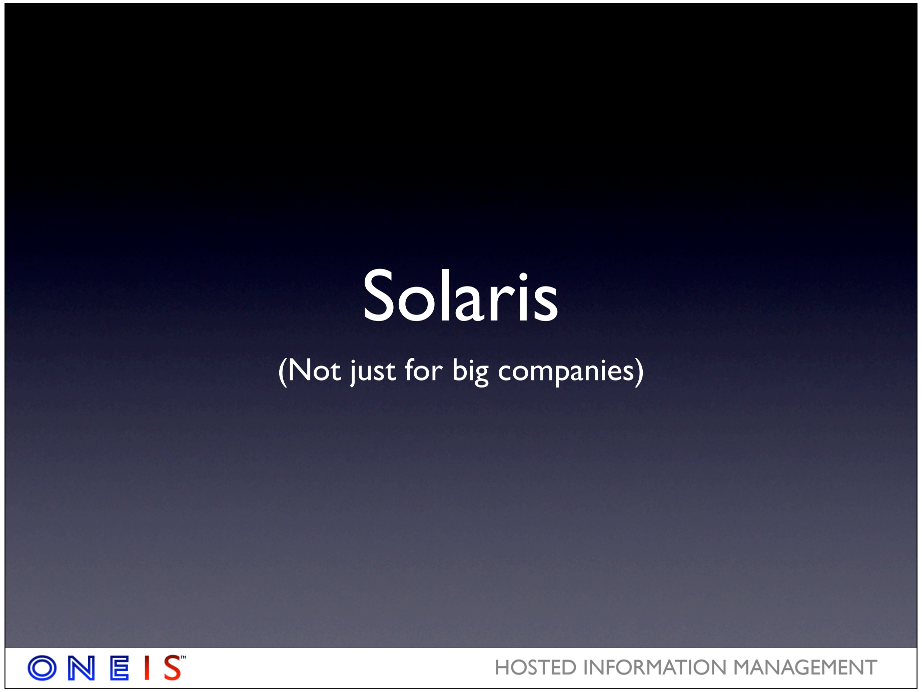
Ben
Development
Sysadmin
...

Jennifer
Information design
Marketing
...

ONEIS™ HOSTED INFORMATION MANAGEMENT

And OneIS is a VERY small company. There’s just two of us. I do everything technical and Jennifer does everything else.

There’s no time to waste, and we’ve got to get the most out of our limited budget.



Solaris

(Not just for big companies)

ONEIS™ HOSTED INFORMATION MANAGEMENT

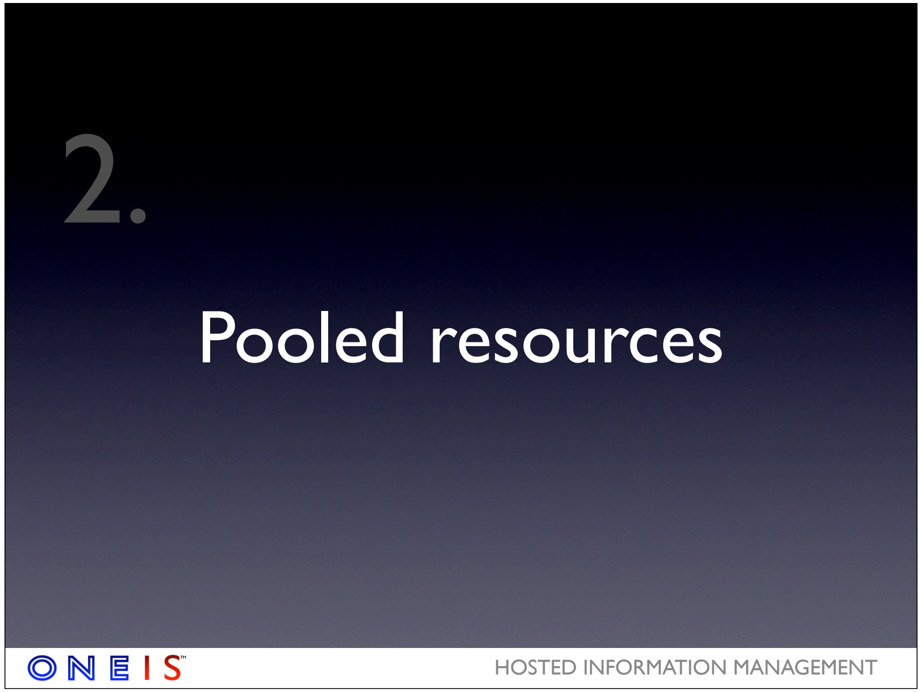
Which is why we chose to run OneIS on Solaris, even though it’s an unusual choice for a small company.
A couple of years ago, I moved to Solaris because I was using OpenBSD, and it wasn’t good with multiple processors. Since many cores is obviously the way things are going, it made sense to choose an OS which could use them effectively.
While I was aware of the funky features of Solaris, it wasn’t the primary motivation. But since then, I’ve been surprised at the positive impact they’ve had in saving time, and getting the most out of our servers.
So, here’s how Solaris has changed the way I work for the better:

1.

Environments:
Dev = Test = Deploy

ONEIS™ HOSTED INFORMATION MANAGEMENT

Firstly, Solaris Zones remove a lot of the hassle from development, testing and deployment.
Since a single physical server can behave like thousands of servers, it’s trivial and almost free to get another virtual server.
So instead of writing code and configuration to cope with a limited number of servers, and dealing with the inevitable differences in environments, you grab as many identical environments as you need and have them all running at the same time.
Our development and testing of the entire application cluster runs on a single server in the office.
Nothing else has such lightweight virtualisation, and it’s made things so much easier and faster.



2.

Pooled resources

ONEIS™ HOSTED INFORMATION MANAGEMENT

Since using ZFS and Zones on Solaris, I’ve been thinking differently about resources like discs and processors. Instead of having a single OS image on a single box, and manually allocating these resources, you take a nice big server and cut out chunks as you see fit, and this can change over time.

So I don’t need to guess in advance how much an application will need over it’s lifetime. I just design the app so it runs in multiple zones, and make sure there’s enough overall resources for everything without worrying about the individual components.

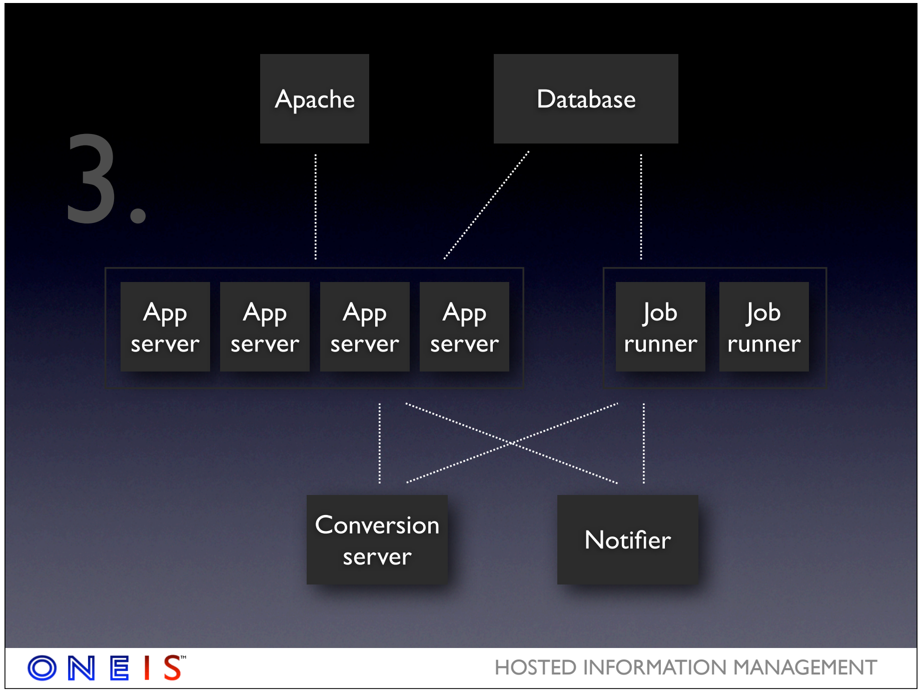
This really simplifies setup and ongoing maintenance, which is great when your sysadmin is also your developer.

3.

Service management built in

ONEIS™ HOSTED INFORMATION MANAGEMENT

There are lots of individual processes in our application, which have to be started in the right order and kept running.
Solaris' SMF does the job very well, and because it's a core OS feature I'm confident that it's well used and well tested, so it'll just work.



3.

Apache

Database

App server

App server

App server

App server

Job runner

Job runner

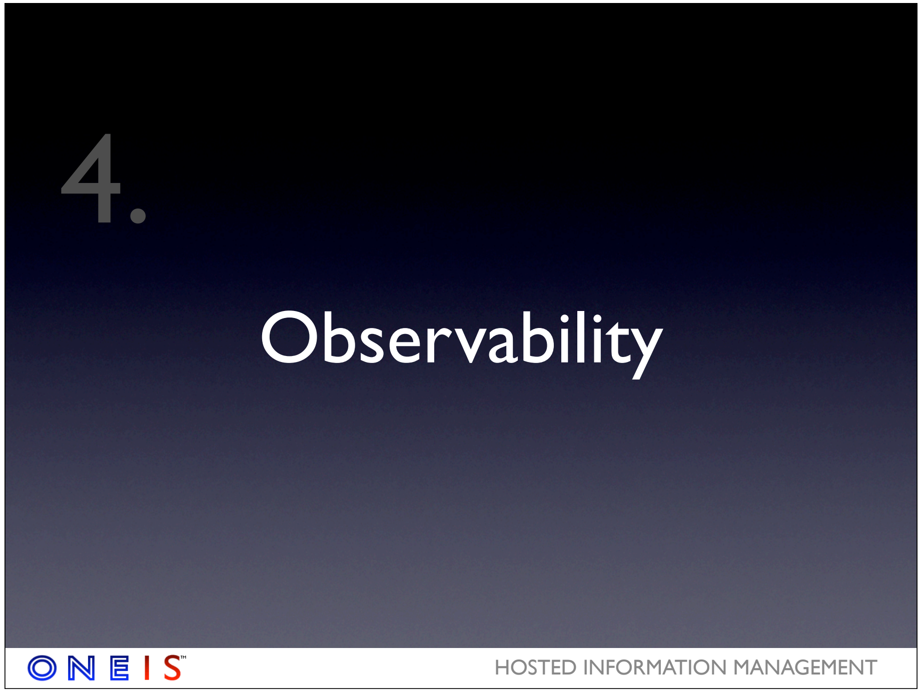
Conversion server

Notifier

ONEIS™

HOSTED INFORMATION MANAGEMENT

Here’s the processes in OneIS as it currently stands, with all the interdependencies shown. With SMF, all the relationships between the components and what to do when something fails is expressed in a few lines of XML. This prompted a new way of thinking about handling those rare failures. Just fail fast and end the process, and let SMF restart everything which needs to be restarted.
That’s potentially thousands of lines of code I didn’t have to write, and some complex testing I didn’t need to do. Yet more time saved!



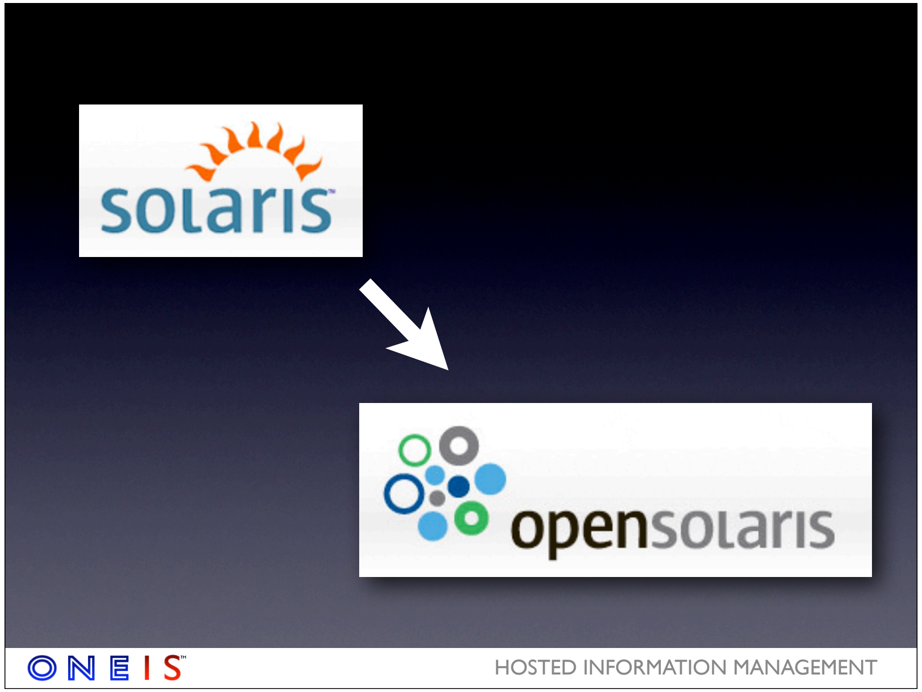
4.

Observability

ONEIS™ HOSTED INFORMATION MANAGEMENT

Optimising and debugging is something you want to get done as quickly as possible because it doesn’t add features. Of course, when you’ve got lots of interrelated processes, it’s not trivial working out what’s going on. But if you could look at the system as a whole, not just single processes, you’d be on to something. And of course, DTrace makes it easy for me to do that.

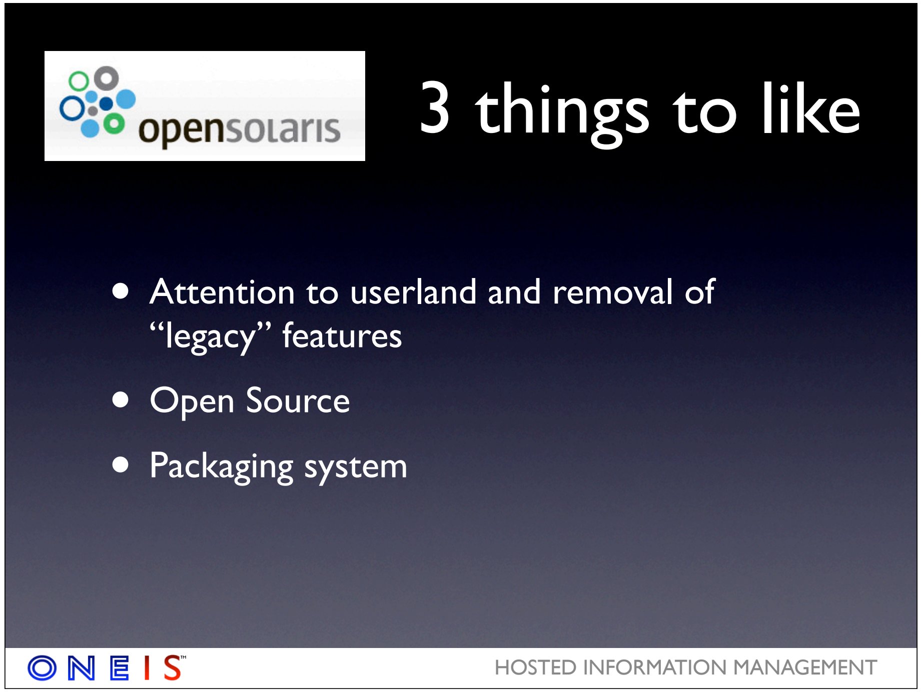
Those demos where Sun people find and resolve problems in seconds? They’re not exaggerating that much.



ONEIS HOSTED INFORMATION MANAGEMENT

OneIS has so far been built on Solaris but I'm hoping to move to OpenSolaris at some point.

When I started with Solaris, OpenSolaris was young, not something you could download and install. It's come a long way since, and moved in a very promising direction.



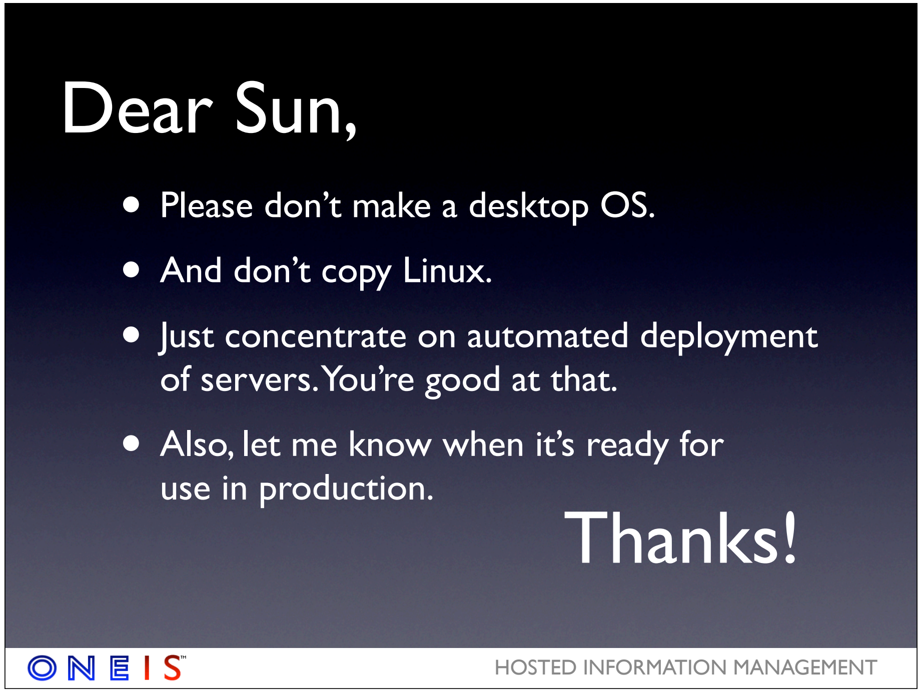
opensolaris

3 things to like

- Attention to userland and removal of “legacy” features
- Open Source
- Packaging system

ONEIS™ HOSTED INFORMATION MANAGEMENT

Here’s my current favourite bits.
Firstly, the ‘unusual’ Solaris userland is given some attention, and the costs of the famed backwards compatibility are being considered carefully. It looks much easier for a newcomer to get to grips with OpenSolaris than it ever was with Solaris.
Secondly, I wouldn’t want to use anything for development where I didn’t have the source to everything, as black boxes are annoying and troublesome. So it’s great that it’s properly open source.
And finally, I’m pleased that SysV packages are going and being replaced by something brand new and Solaris specific. Adopting any existing packaging system would only have clashed with all the lovely bits I’ve been talking about.



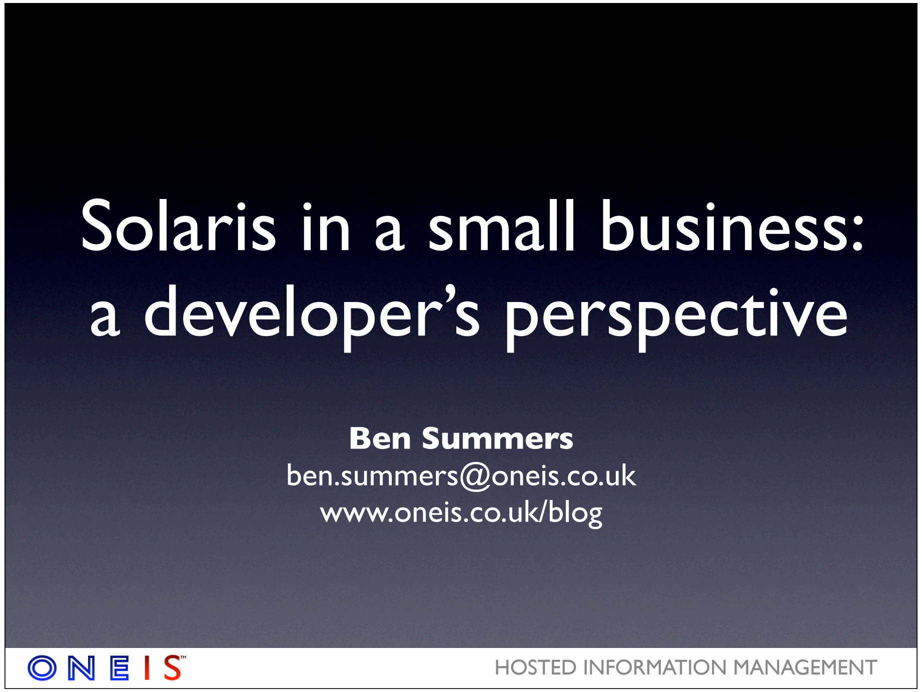
Dear Sun,

- Please don’t make a desktop OS.
- And don’t copy Linux.
- Just concentrate on automated deployment of servers. You’re good at that.
- Also, let me know when it’s ready for use in production.

Thanks!

ONEIS™ HOSTED INFORMATION MANAGEMENT

Everyone likes to tell Sun what they should do. So here’s what I think.
Don’t put too much effort into the desktop. Open source desktops are just poor copies of Windows.
OpenSolaris on the desktop would just be a copy of every other UNIX desktop. Leave innovation there to Apple.
Don’t copy Linux, but keep up the good work and concentrate on making it wonderful on the server. Make the packaging system a easy way to install the OS and my application and everything it needs, and then manage upgrades.
Finally, be clear about when it’s production ready.



Solaris in a small business: a developer's perspective

Ben Summers
ben.summers@oneis.co.uk
www.oneis.co.uk/blog

ONEIS™ HOSTED INFORMATION MANAGEMENT

The decision to develop OneIS on Solaris has had some rather pleasant benefits, changed the way I design and develop software, and has helped me to do an awful lot with limited time and resources.
Thank you.