Graham Gilbert / Brett Demetris / 20180220

Curing operating system blindness

A long time ago...

In Dog Years

- macOS is the only first class citizen
- Commercial management tool
- Unmanaged Windows VM available
- What’s security?

Not so long ago...

We found our way

- macOS is our single focus
- Munki is implemented
  - Established Patching SLA
- Source control is implemented
  - Global Visibility
  - Code Review
  - Cross Team Function
- Strong focus on Security

What are we trying to achieve?™

- Keep Airbnb's, our host's and our guest's information secure
- Enable Airbnb employees to be productive

What tools do we have?

What tools do we have?

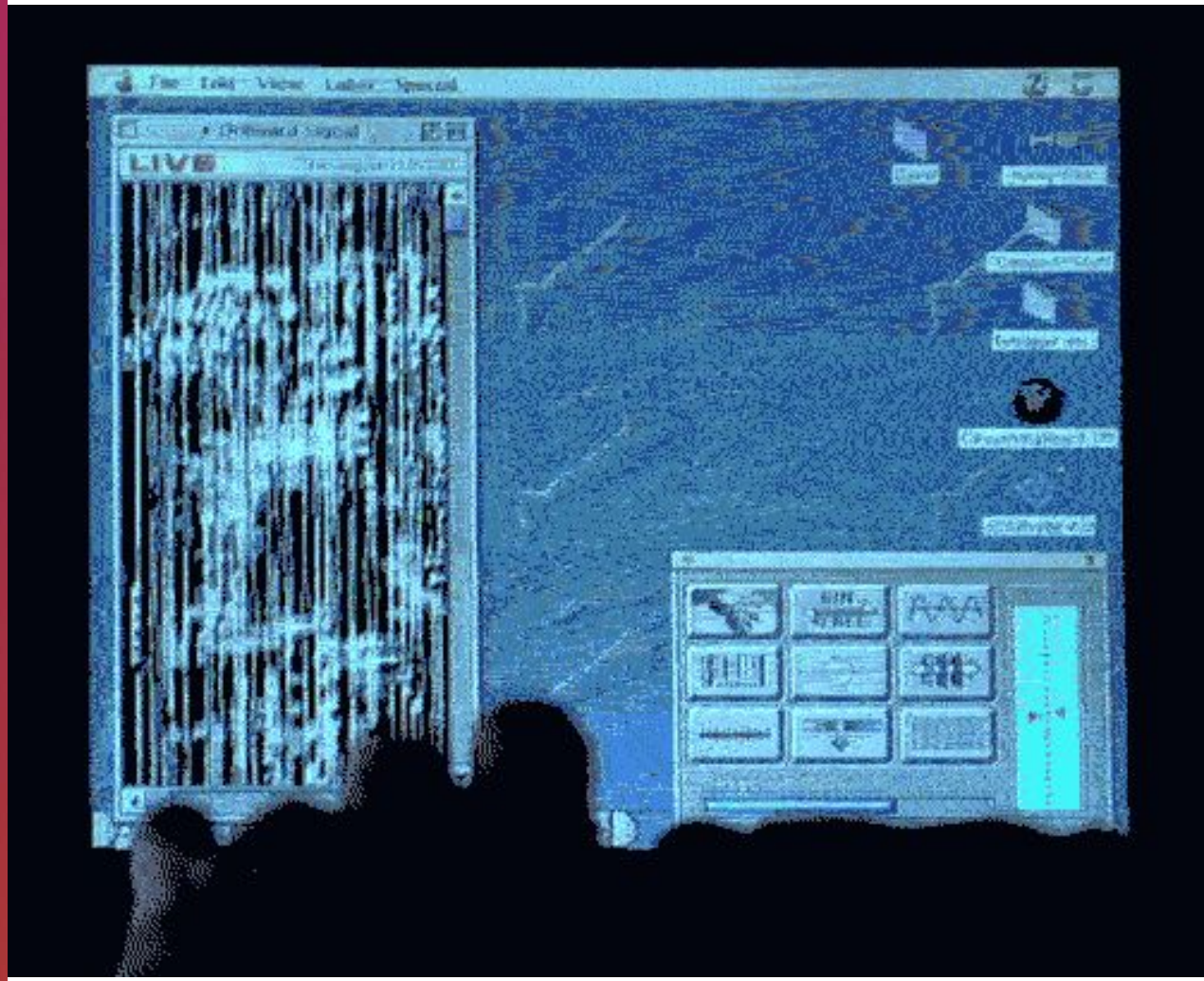
- **Munki** Software updates
- **Crypt** FileVault key escrow
- **Sal** Reporting
- **Puppet** Configuration management
- **microMDM** DEP / Bootstrapping

And then we bought a company

“Hey, guys…. How do you feel about managing Windows?”

HORRORFIXXX

“Just give them a Mac”



“Just give them a Mac”

- We don’t have processes to manage another platform
- We don’t have time to manage another platform
- We don’t *want* to manage another platform

Their entire product was built on a Microsoft stack

Financey software

Are we the baddies?

Be a host

We need to support Windows

Let's Yabba-Daba
THE FLINTSTONES
STONE AGE SMACKDOWN!

macOS

Patching (Munki)
Configuration (Puppet)
FileVault (Crypt)
Reporting (Sal)
MDM (microMDM)

Windows

Patching (??)
Configuration (Puppet)
BitLocker (??)
Reporting (??)
MDM (??)

How do we do it on a mac?

Install a Certificate

```
define certificates::root (
  String $source,
  String $sha1,
  String $trust_level = 'trustRoot',
  )
```

A Certificate is pretty simple on a mac

It takes a source, a sha1, and a trust level

```
certificates::root { 'My Internal Cert':
    source      => "puppet:///modules/internal_certs/my_cert.crt",
    sha1        => '5bc875fa06f5105b36f0913f199ce98e67d24895',
    trust_level => 'trustRoot'
}
```

How does it look on Windows?

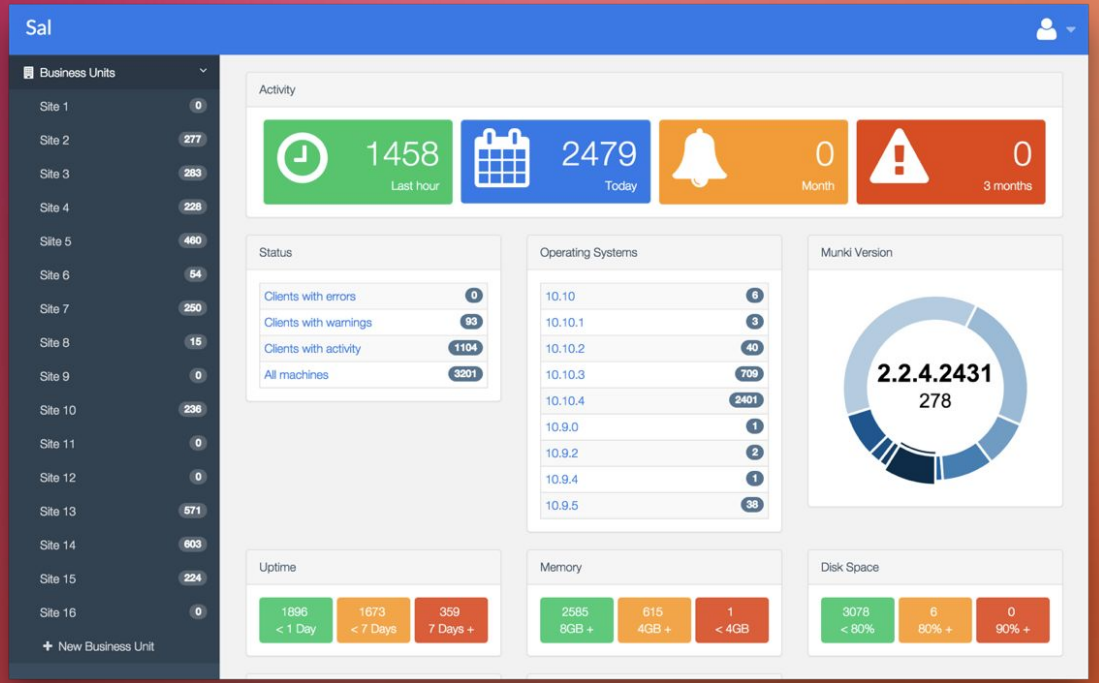
What did you expect?

```
sslcertificate { "mycert.crt":
  name       => 'mycert.crt',
  password   => 'password123',
  location   => "${::facts['puppet_vardir']}/certificates",
  thumbprint => upcase($sha1)
}
```

- Windows and Mac have the same implementation on the surface
- Providers handle the OS specifics under the hood
- The properties of a certificate don’t change between Operating Systems

Stop offering the OS
Start offering the service

IF A CLIENT DOESN'T CHECK IN
DOES IT EXIST?


Sal
Business Units
Site 1 0
Site 2 277
Site 3 283
Site 4 228
Site 5 460
Site 6 54
Site 7 250
Site 8 15
Site 9 0
Site 10 236
Site 11 0
Site 12 0
Site 13 571
Site 14 603
Site 15 224
Site 16 0
New Business Unit
Activity
1458 Last hour
2479 Today
0 Month
0 3 months
Status
Clients with errors 0
Clients with warnings 93
Clients with activity 1104
All machines 3201
Operating Systems
10.10 6
10.10.1 3
10.10.2 40
10.10.3 709
10.10.4 2401
10.9.0 1
10.9.2 2
10.9.4 1
10.9.5 38
Munki Version
2.2.4.2431
278
Uptime
1896 < 1 Day
1673 < 7 Days
359 7 Days +
Memory
2585 8GB +
615 4GB +
1 < 4GB
Disk Space
3078 < 80%
6 80% +
0 90% +

Gosal

A Windows Client for Sal

- Powershell exec
- Go Struct
- POST Request

github.com/airbnb/gosal

But... We have a third operating system

Chrome OS

Why Chrome OS?

- No more imaging / provisioning
- Built in patching
- Verified Boot
- Decent devices are really cheap

But you can only do what Google lets you, right?

among
That

Apps

Extensions

Learn Basics
Getting Started Tutorial
Samples
Develop Extensions
Distribute Extensions
Chrome Platform APIs
JavaScript APIs
Manifest File Format
Web APIs
Permission Warnings
Optional Permissions
Match Patterns
Help

Native Client

Store

JavaScript APIs

Chrome provides extensions with many special-purpose APIs like `chrome.runtime` and `chrome.alarms`.

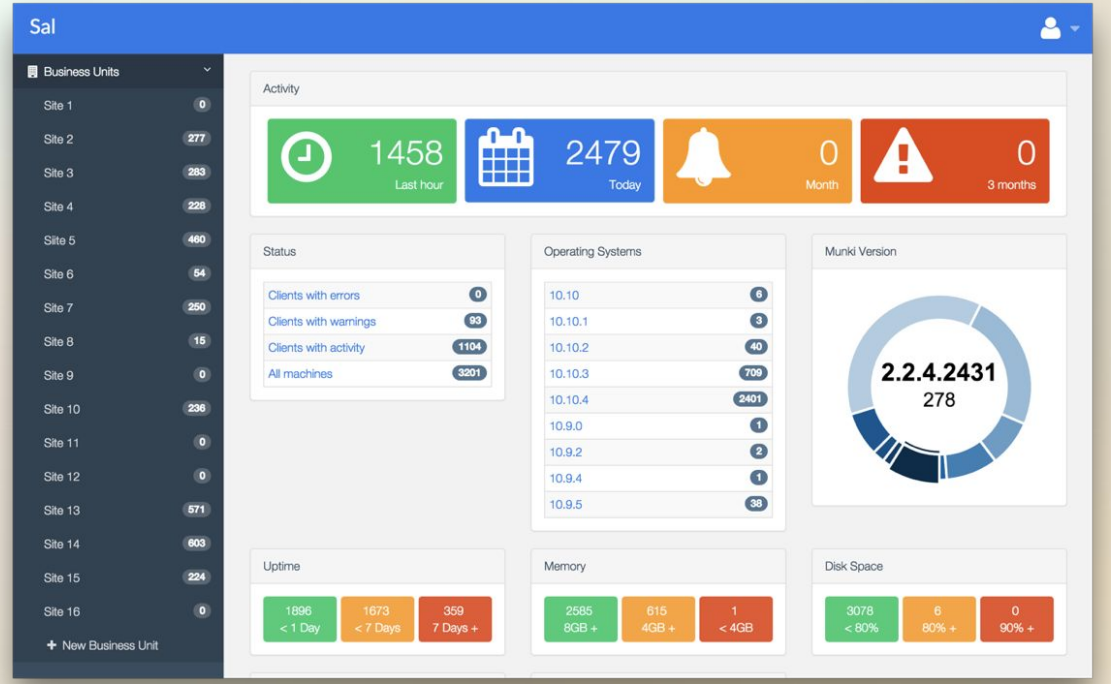
Contents

Stable APIs
Beta APIs
Dev APIs
Experimental APIs
API conventions

Stable APIs

Name	Description	Since
accessibilityFeatures	Use the `chrome.accessibilityFeatures` API to manage Chrome's accessibility features. This API relies on the **ChromeSetting prototype of the type API** for getting and setting individual accessibility features. In order to get feature states the extension must request `accessibilityFeatures.read` permission. For modifying feature state, the extension needs `accessibilityFeatures.modify` permission. Note that `accessibilityFeatures.modify` does not imply `accessibilityFeatures.read` permission.	37
alarms	Use the `chrome.alarms` API to schedule code to run periodically or at a specified time in the future.	22
bookmarks	Use the `chrome.bookmarks` API to create, organize, and otherwise manipulate bookmarks. Also see **Override Pages**, which you can use to create a custom Bookmark Manager page.	19
browserAction	Use browser actions to put icons in the main Google Chrome toolbar, to the right of the address bar. In addition to its **icon**, a browser action can also have a **tooltip**, a **badge**, and a **popup**.	19

IF A CLIENT DOESN'T CHECK IN
DOES IT EXIST?


Sal
Business Units
Site 1 0
Site 2 277
Site 3 283
Site 4 228
Site 5 460
Site 6 54
Site 7 250
Site 8 15
Site 9 0
Site 10 236
Site 11 0
Site 12 0
Site 13 571
Site 14 603
Site 15 224
Site 16 0
New Business Unit
Activity
1458 Last hour
2479 Today
0 Month
0 3 months
Status
Clients with errors 0
Clients with warnings 93
Clients with activity 1104
All machines 3201
Operating Systems
10.10 6
10.10.1 3
10.10.2 40
10.10.3 709
10.10.4 2401
10.9.0 1
10.9.2 2
10.9.4 1
10.9.5 38
Munki Version
2.2.4.2431
278
Uptime
1896 < 1 Day
1673 < 7 Days
359 7 Days +
Memory
2585 8GB +
615 4GB +
1 < 4GB
Disk Space
3078 < 80%
6 80% +
0 90% +

Sal

Search...

- Advanced Search
- Chrome OS Devices
- Overview
- Facter
- Conditions
- Application Inventory
- Computer

Chrome OS Device

Report :: 2018-02-11 22:14

Pending Installs

Notifications

0 Errors

0 Warnings

0 Pending Installs

Machine Details

Model:	Google Pixelbook
Processor:	Intel(R) Core(TM) i3-5005U CPU 2.00GHz
RAM:	8 GB
Serial:	A0200B86-AAA0-00A0-0A0A-AAAA00000
OS version:	64.0.3282.144
Deploy Status:	Deployed

Management Tools

Sal

Search...

Advanced Search

Back

Export CSV

Application Inventory for Machine: Chromebook

Show 25 entries

Search:

Name	Bundle ID	Bundle Name	Install Count
1Password X (beta)	khgocmkkpikpnmmkgmdnfckapcdkgfaf		1
Chromebook Recovery Utility	jndclpdbaamdhonoechobihbbiimdgai		1
ChromeSal	ohnahlfkoiaeknjhhcnhcjgpemccoklg		1
Cisco WebEx App for Chrome	fceempjejlfaadkgdacpfhhekndlcjl		1
Cisco WebEx Extension	jlhmfgmfgeifomenelglieieghnjghma		1
GlobalProtect	nicidmbokaedpmoegdbcebhnchpegcdc		1
Google Docs	fgbakocaamckmahiegoajipleckepgpm		1
Google Drawings	comidegkhoojfbjhioinljjilfjolcgj		1
Google Forms	kdnjchdofihcohancegjjogehalhmkba		1
Google My Maps	bpfmlkhmlbehehljkbemjlideclmhlpb		1
Google Play	komhbcfkdcgmcdoenjcjheifdiabikfi		1
Google Sheets	eeipohcoolhaigpjlfknjbbedjgkpenc		1
Google Sites	phhmhpgbmkjhhocndpfbhpbakkdagdil		1
Google Slides	holaiadndlekbinndlbnaiddjlcpfobi		1
Google Verified Access by Duo	lojeokmpinkpmpbakfkfpgfhpapbgdnd		1
Lucidchart Diagrams	oilengkmkbgjlbiopfmekaaffieepokm		1

ChromeSal

- graham.at/chromesal
- github.com/salopensource/chromesal
- Requires Sal 3.3.0

What’s next?

Package management for Windows

Windows package management

- An army of helpdesk people to install things by hand
- Chocolatey Enterprise
- Munki for Windows?

Managing Chrome OS is GUI based

More information

Questions?

graham.at/movember