

Crowd Sourcing Access Control in the Cloud

Marc Stiegler
HP Labs



Session ID: CLD-204

Session Classification: Advanced

RSA CONFERENCE 2012

More choices? Or Better Ones?

- A funny thing happened on the way to crowd sourcing access control
- The choices:
 - Submit a poor paper
 - Share credentials
 - Ask HP-IT to modify policy and software in 4 hours

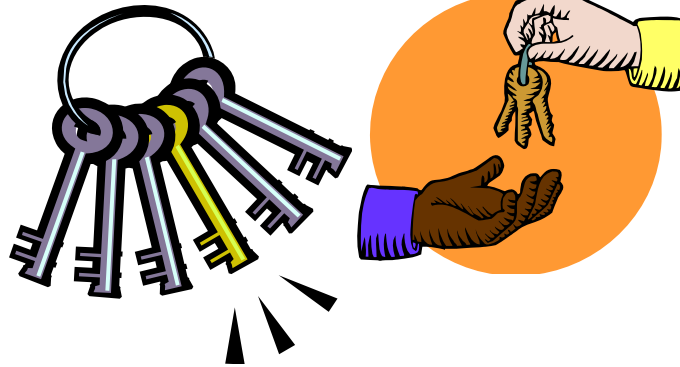


Better Choices: Rich Sharing

Dynamic



Attenuated



Chained



Cross Domain



Accountable

Recomposable



Why is Rich Sharing So Hard?

- Authentication Based Access Control
 - Authenticate at time of access
 - Attenuated chained delegation: banned by ACLs

```
marcstgr@marcstgr-8530w:/etc$ chmod +r adduser.conf
chmod: changing permissions of `adduser.conf': Operation not permitted
marcstgr@marcstgr-8530w:/etc$ █
```

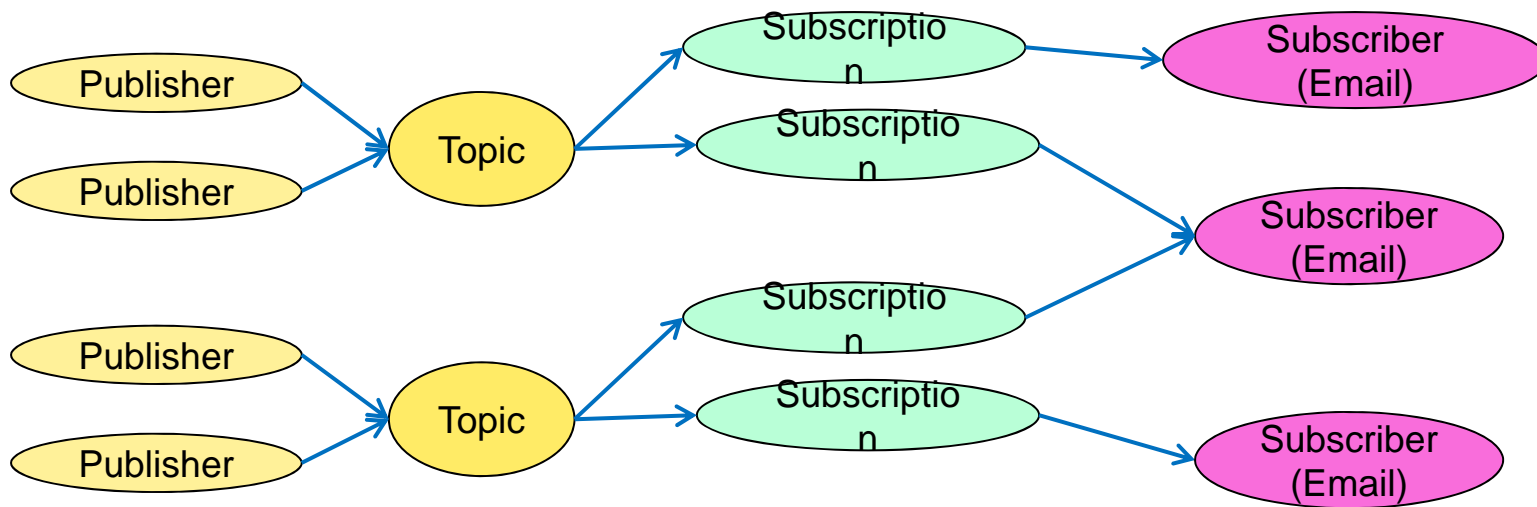
- Authorization Based Access Control (ZBAC)
 - Authenticate at time of grant, like car keys
 - As easy as unguessable urls: self-authorizing links
 - Webkeys, self authorizing links on steroids:

<https://sha-256-hl6w2x74ixy6pi5n.yurl.net:4445/-/cookieshop/#s=gijlimpujtfcho>

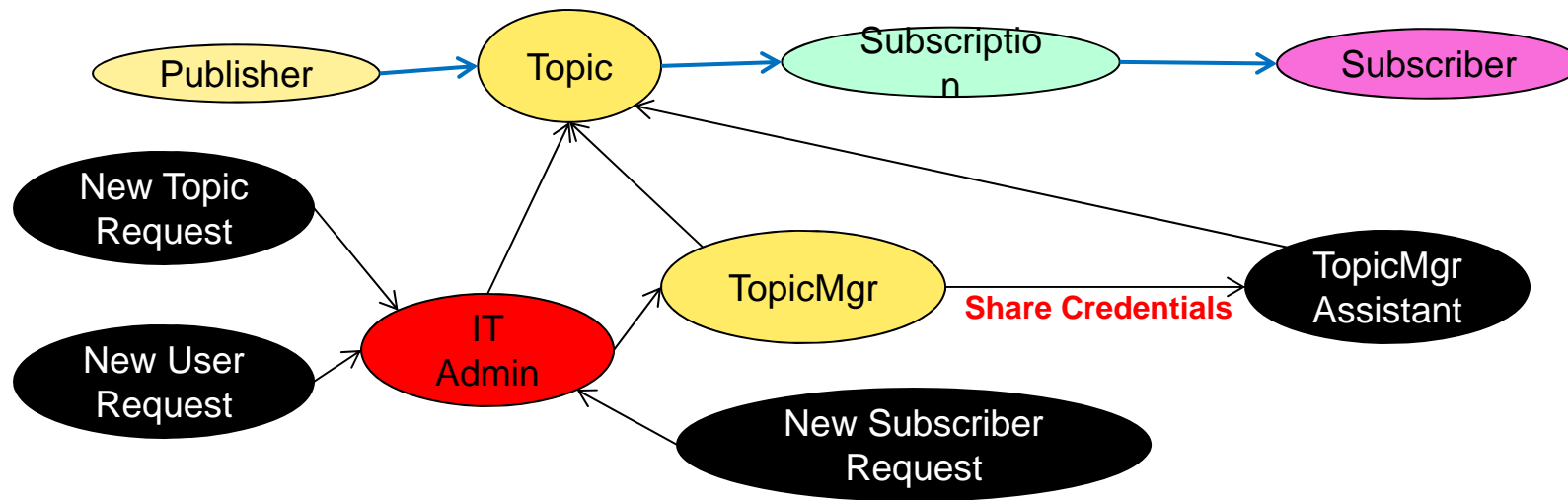


Cloud Example: Amazon SNS

- A Publication/Subscription System
- Traditional Access Control



Computation Scales; Access Control Fails



- Minor Problems:
 - Credential sharing increases attack surface, decreases accountability
 - User work quality falls, institution becomes sluggish, revenues fall, IT budget cut
- *Serious Problem: IT cost grows faster than linearly*

Stick it to the Users



2 Strategies for Sticking Users

- Have a Nice Day Strategy (increasingly popular)

From: noreply@corpIT.com
To: youLoser
Subject: You are in Big Trouble

You have violated corporate policy. Please rectify using this link:
<https://corporatePageDisguisedAsPhishingSite.com>
If you have any questions, do not even try to contact us.

Have a nice day.

(P.S. If this were a self-authorizing link, it would be fine. More later)

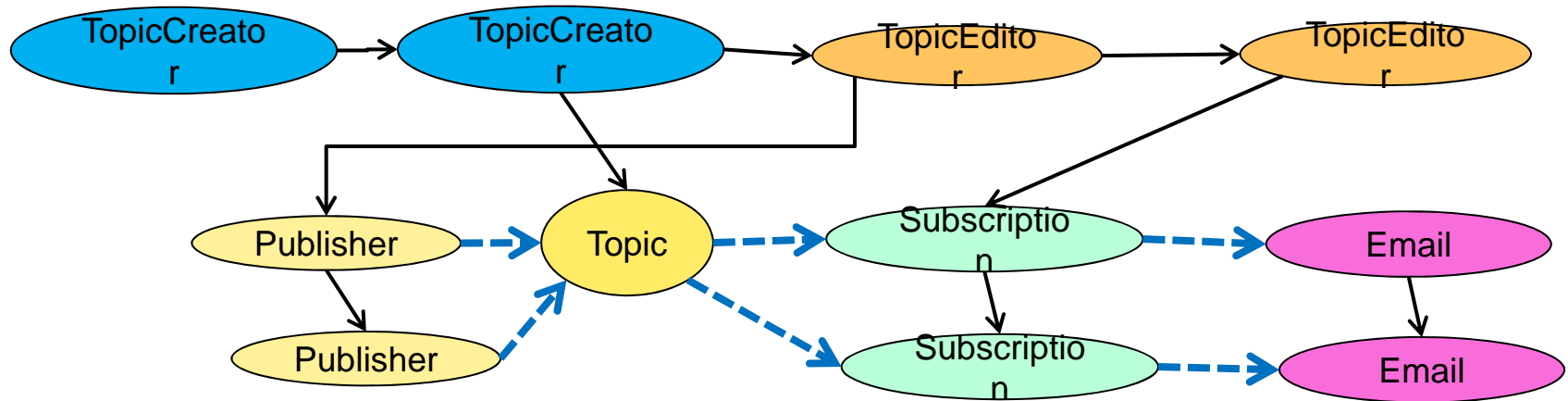
- Can we not do better?



Crowd Sourcing Strategy

- Give the Users both *tools* and *responsibility*.
 - Like the switch from telephone operators to direct dial
 - Users loved having to do the extra work. Ah, the freedom!
 - What tools do they need?
 - Attenuate and share. Attenuate again and share again
 - Cross domains without IT help
 - Keep track of who did what
- In Other Words
 - Attenuated chained accountable delegation across admin domains ... Rich Sharing!
- Aha! Rich Sharing *enables* Crowd Sourcing



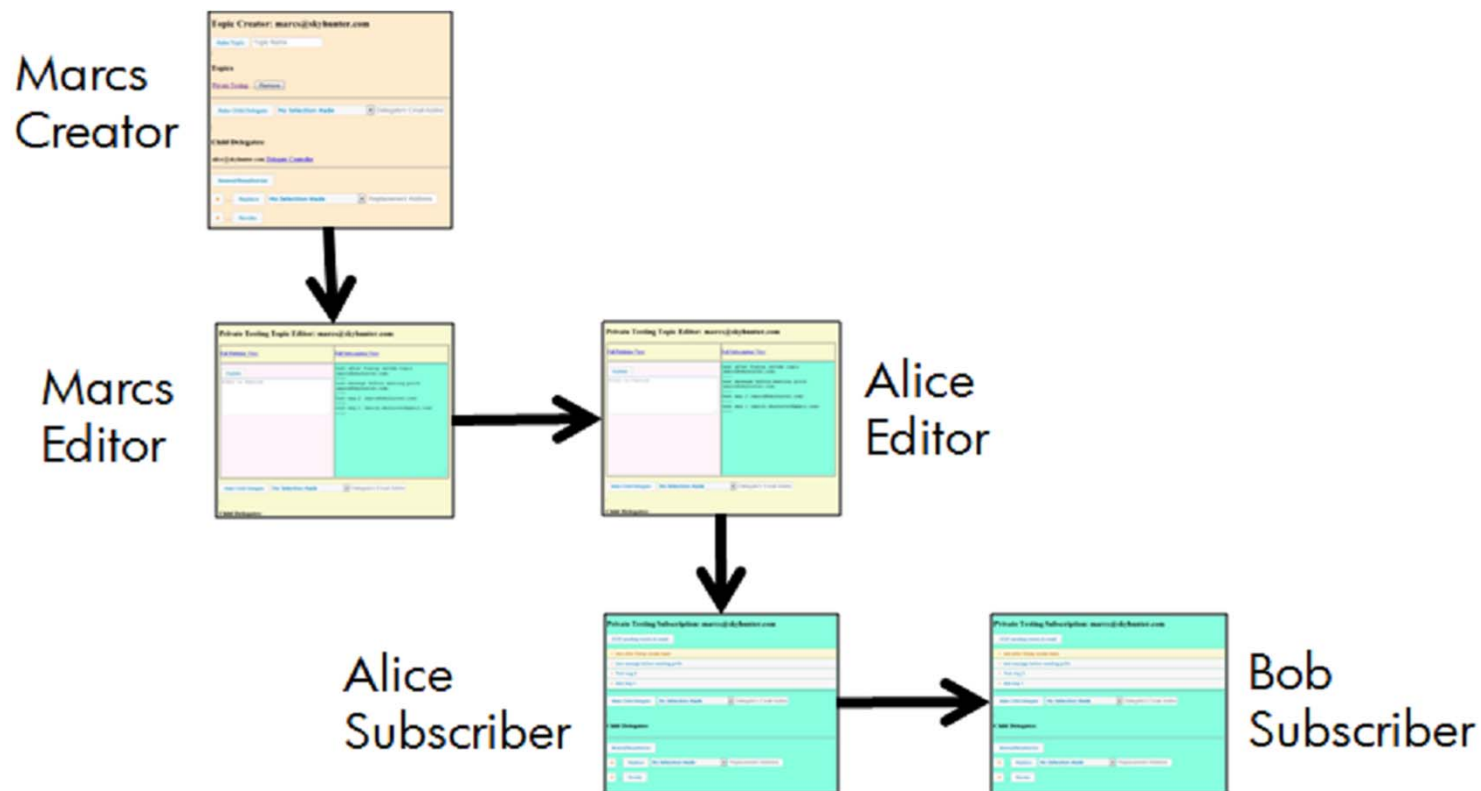
Crowd Source Pub/Sub with Rich Sharing



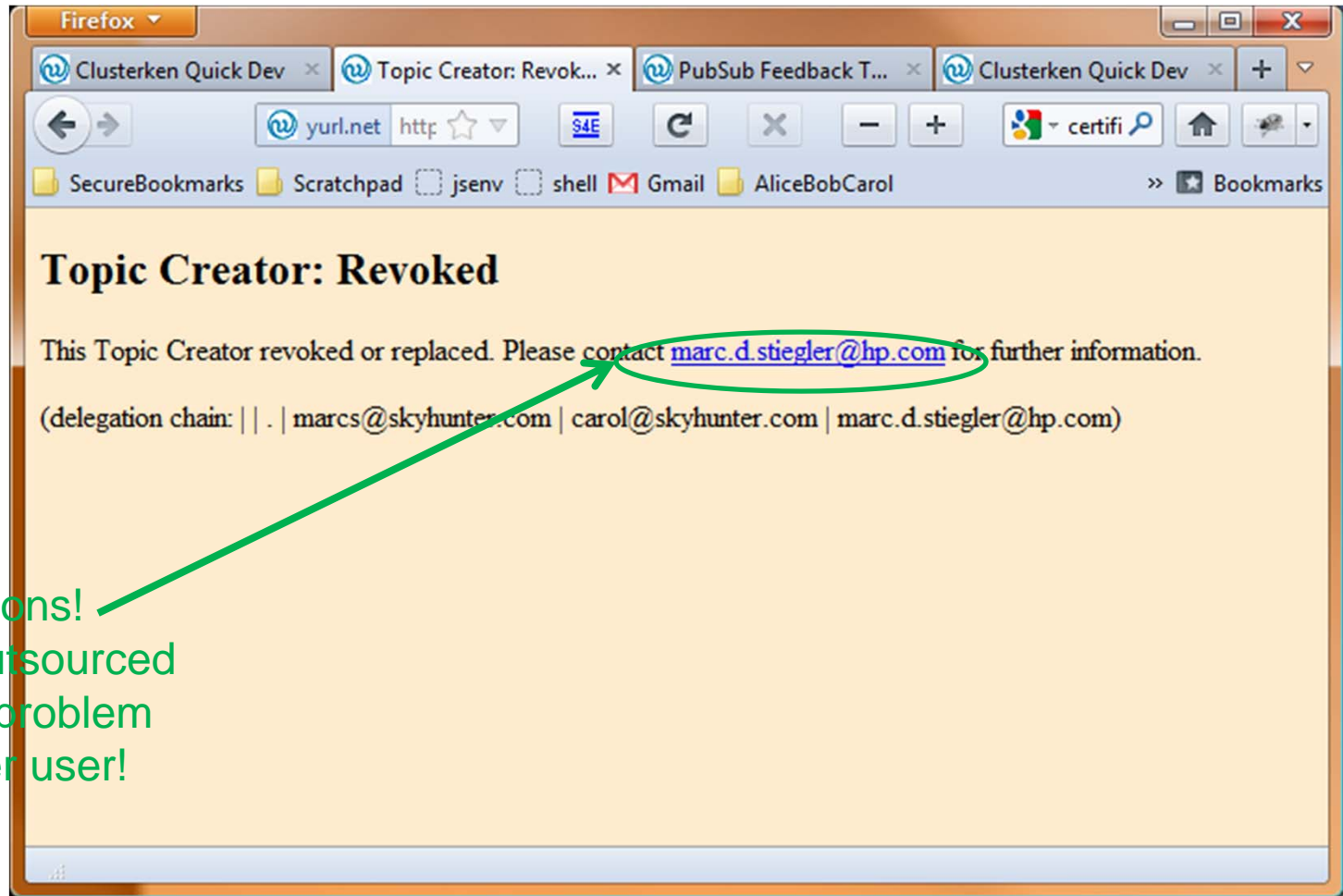
Black arrows  flow of creation/delegation
Blue arrows  flow of data

Basic Pub/Sub Demo

- To Sing Along
 - Go to skyhunter.com/pubshare submit your email address, receive links.



The Essence of Crowd Sourcing



Congratulations!
You have outsourced
solving this problem
... to another user!



The Gotcha that
every RSA audience
will spot



Interlude

- Does anyone have a Cherry Coke I could buy?



Revocation...or Replacement? 3 cases

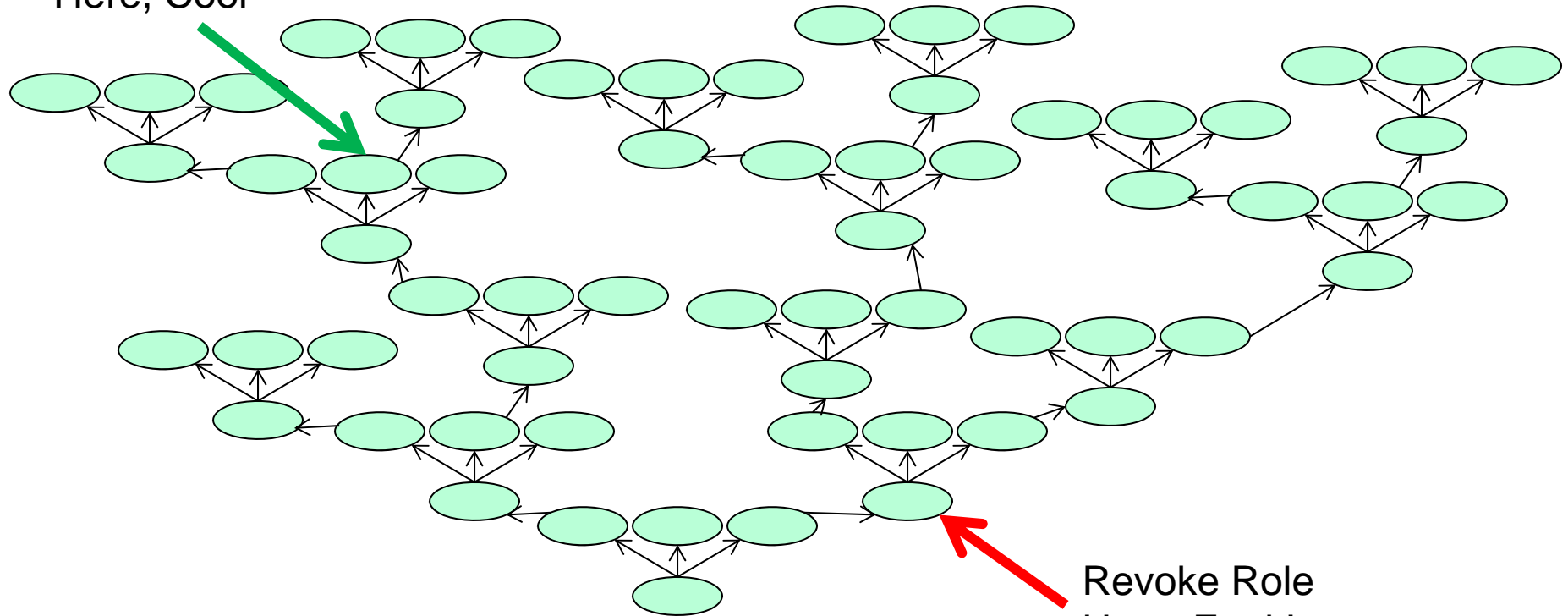
- SubProject comes and goes
 - Revoke all the related authorities
- Delegator of many topics just changed jobs
 - He can no longer fulfill this role
 - Replace
- Delegator of many topics fired
 - Replace with extreme prejudice



Deep Trees and Sock Puppets

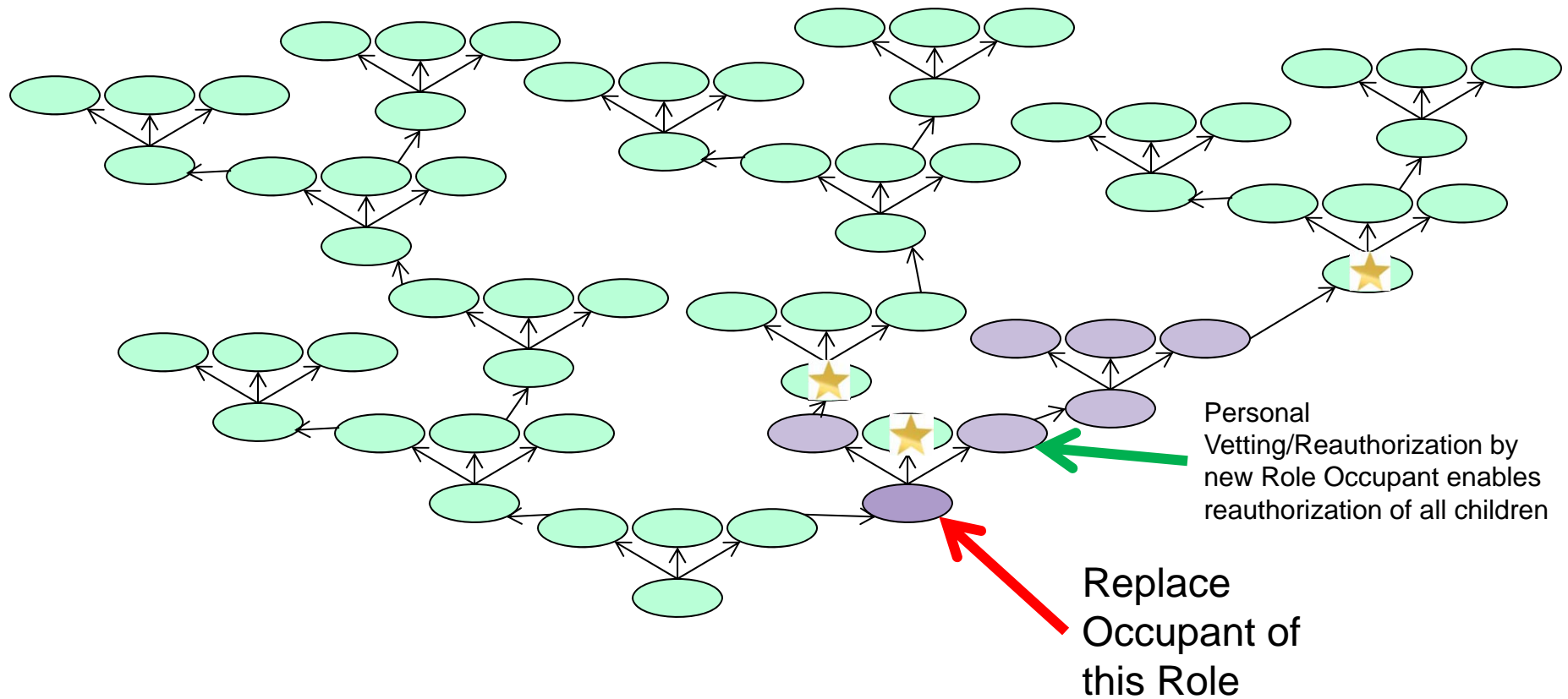
Work mostly done near top,
Administration mostly done near bottom

Revoke Project
Here, Cool



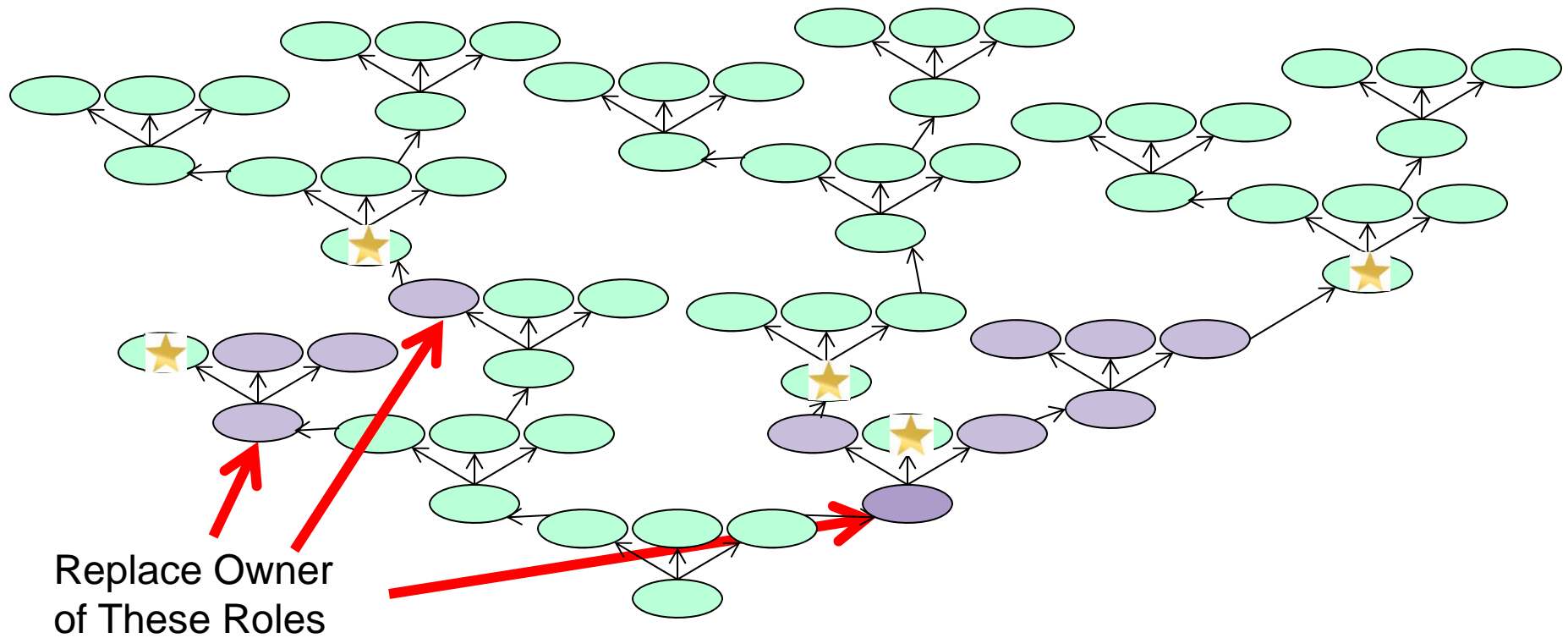
Revoke Role
Here, Eeek!

Replace: Introduction of Vetted Email Addresses



Replace with Extreme Prejudice

Extreme Prejudice still needs IT Admin actions:
Remove email address from vetted list,
Start at Root, Search and Replace Everywhere



Demo with Vetted Email Addresses

- Again, play with this yourself



Building Your Own Crowd Sourced Access App



A Rich Sharing, Crowd Sourcing Hello Cloud

- Richly shared event log
- Using Clusterken
 - Open source cluster framework for Java and Scala
 - Object-level message passing
 - Webkeys supply OO referential integrity
 - Access control graph == Object reference graph
- Code nearly what you would write for a sequential single node program with no access control, yet it scales over the cloud with strong security properties.



Clusterken Quick Dev

The screenshot shows the 'Clusterken Quick Dev' web application running in a Firefox browser. The browser's address bar shows the URL: `http://yurl.net/~sha-256-mnj4yjwqtwi2tjwn.yurl.net:4450/~quickdev/#s=i2physy55giekd`. The application interface includes a header with the title 'Clusterken Quick Dev', a navigation bar with tabs for 'Clusterken Quick Dev', 'PubShare Discussion Forum Topic: m...', 'PubSub Feedback Topic: marc.d.stieg...', and 'Clusterken Quick Dev'. The main content area features a code editor displaying the following Java code:

```
48 */
49 public interface RichSharedLog extends Clustered {
50
51     public Ok append(String text);
52     public Promise<String> text();
53     public Ok revokeChild(String petname);
54     /**
55      *
56      * @param petname
57      * @param authorization Uses the root of the log tree, which all richsharedlog nodes have
58      * but no other objects or people have, as proof that the invoker of this method
59      * is allowed to revoke.
60      * @return
61      */
62     public Ok revoke(RichSharedLog authorization);
63     public Promise<Boolean> revoked();
64     public RichSharedLog makeDelegate(String petname, boolean readOnly);
65     public Promise<String> delegationPath();
66     public Promise<ConstMap<String, RichSharedLog>> delegates();
67     public RichSharedLog init(String name, String parentPath, RichSharedLog root,
68         boolean isReadOnly);
69     public Promise<Boolean> yesReadOnly();
70
71     class X extends Clustered.X implements RichSharedLog, Serializable {
72         private static final long serialVersionUID = 1L;
73
74         private String text = "";
75         private String myName = ".";
76         private String parentPath = "";
77         private boolean revoked = false;
78         private boolean isReadOnly = false;
79         private RichSharedLog root = null; //the root itself will have a null value here
80         private ChrPrMan<String, RichSharedLog> delegates =
```

The sidebar on the right contains a 'Local Script' section with buttons for 'Clone QuickDev' and 'Help Links...'. Below that is an 'Enlarge' button. The 'FarFile Directory' section shows a list of files and folders, with 'RichSharedLog.java' selected. The directory path is `/home/marcstgr/servers/alanPubShare/clusterApps/src/com/hp/clusterken/samples`. The file list includes:

- Even.java | 1777
- EvenDemo.java | 1489
- EvenTail.java | 3664
- HelloCluster.java | 3168
- HelloWorld.java | 1528
- Odd.java | 1850
- PurseX.java | 1918
- RandGenX.java | 1309
- RichSharedLog.java | 6821**
- TestValidator.java | 2188
- XRec.java | 1429
- package-info.java | 218

Buttons for 'Up', 'Copy', and 'Delete' are visible. At the bottom of the sidebar, there are 'New Dir' and 'New File' buttons. The main content area also includes a 'Save Source' button and 'Build Project' and 'Propagate' buttons. The status bar at the bottom of the code editor shows 'Position: Ln 166, Ch 1' and 'Total: Ln 166, Ch 6656'.



Crowd Sourcing Next Steps



Quick Applications

- Warmup exercises for fully crowd sourced systems
- For one existing system,
 - implement self-authorizing links as a selectable alternative that skips the login/username/password page.
 - This is not bleeding edge. See Google Docs, YouTube
- For systems that send out alert emails with embedded links to login pages: use self authorizing links instead to prevent phishing
 - This is not bleeding edge either. IEEE renewals, IEEE Software submitted paper reviews
- For one future system, spec rich sharing
 - Make explicit req, chains of attenuated delegations!



Conclusion

- What is the proper role of IT?



*If the Apocalypse comes,
keep me**

- Stick it to your users by crowd sourcing access control to them
- Accept their thanks

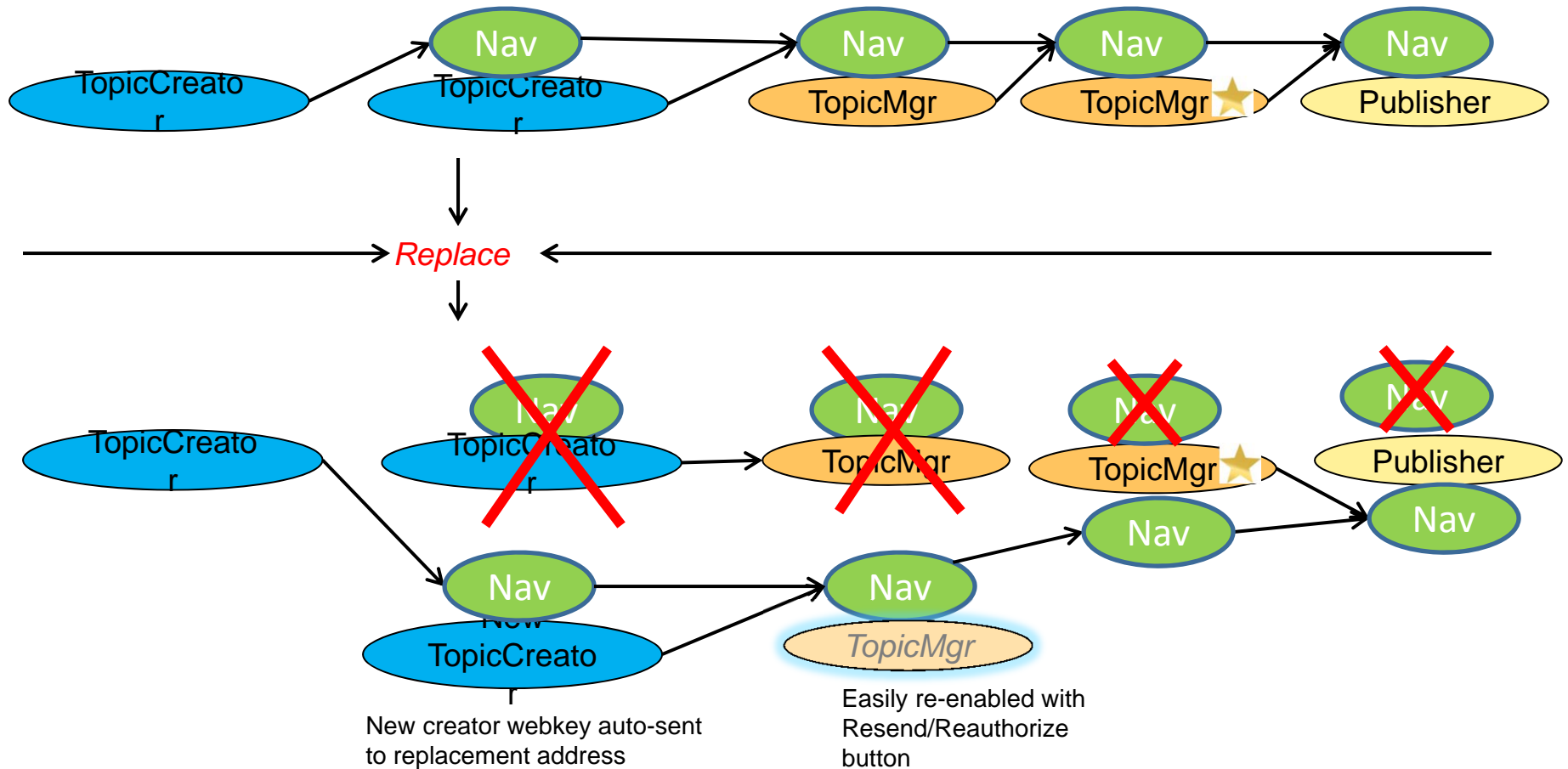
**Buffy the Vampire Slayer, S1, E5,
"Never Kill a Boy on the First Date"*



Backups



Replace Operation



Algorithm essence

- Algorithm's essence
 - When replacing a delegator
 - If delegatee is vetted
 - Do Not replace him
 - Else replace delegatee too
 - Requires manual re-authorization by a delegator/ancestor
 - Recurse to leaves of the tree

