

Session 100

Single Sign-on Integration With PKI

Dan Norris

norris@celeritas.com

Senior Consultant

Celeritas Technologies, LLC



Copyright © 2004 Celeritas Technologies, L.L.C.

This work was created by Celeritas Technologies, L.L.C. (“Creator”). This work and all rights therein and thereto, including copyright rights and/or patent rights, are owned by Creator and/or another entity (collectively, “Owner”). This shall serve as notice of such ownership as of the date of this and associated files or subject matter, if any, as depicted above and/or as depicted with an electronic file date stamp and/or any versions thereof and their associated dates, if any.

This work may not be reproduced for any purpose, distributed, modified, reverse-engineered, stored in a retrieval system, transmitted, used, made, offered for sale, or sold, in whole or part, in any form or by any means, electronic, mechanical, audio, photocopying, recording, or otherwise, without the prior written permission of Owner.

This work may not be exported unless in compliance with the applicable technology export laws. While this information is presented in good faith and believed to be accurate, Creator does not guarantee satisfactory or any results from reliance upon such information.

Creator reserves the right, without notice, to alter or improve the designs, specifications, creations, or works of the subject matter herein. Nothing herein is to be construed as a warranty or guarantee, express or implied, against infringement, or regarding performance, merchantability, fitness, or any other matter with respect to products, processes, or any other subject matter herein, and such warranties and guaranties are expressly disclaimed. Nothing herein is to be construed as a recommendation to use any product or process in conflict with any third party rights in any intellectual property.

All products, languages, or trademarked names that are mentioned in this work are acknowledged to be the proprietary property of the respective owner.



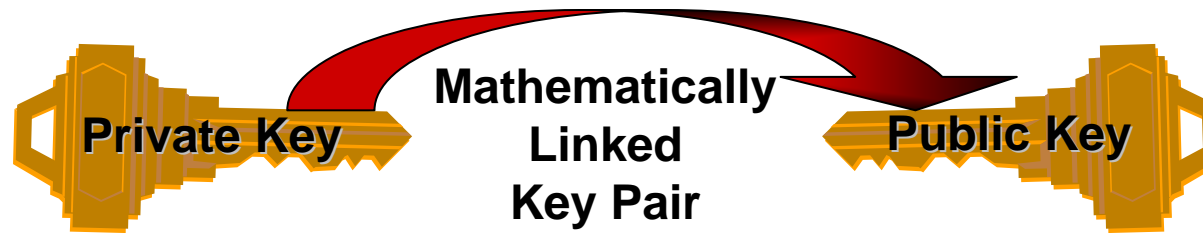
IOUG RAC SIG Events

- Today, 12 noon, Room 709: Expert presentation “Workload Distribution in a RAC Environment”
- Tomorrow (Tuesday), 12 noon, Room 709: RAC SIG Roundtable—”Stump Your Peers”
- Lunch provided both days at Room 709

Agenda

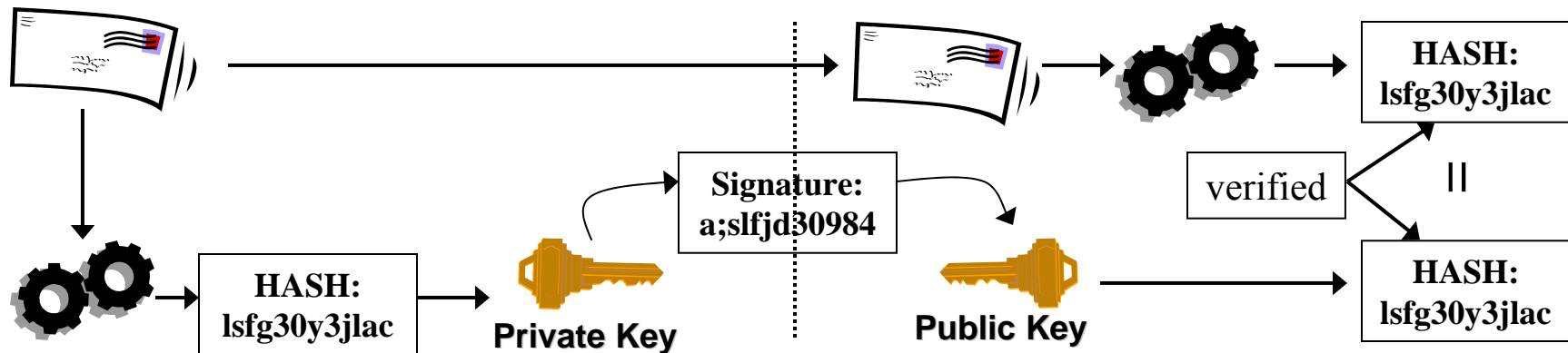
- ▶ What PKI is and is not
 - 9iAS SSO overview
 - 9iAS SSO with digital certificates
 - 9iAS SSO third-party integration point
 - Web-based PKI sign-on
 - Steps to integrate 9iAS SSO with web-based PKI sign-on
 - Q & A

What PKI Is (and Is Not)



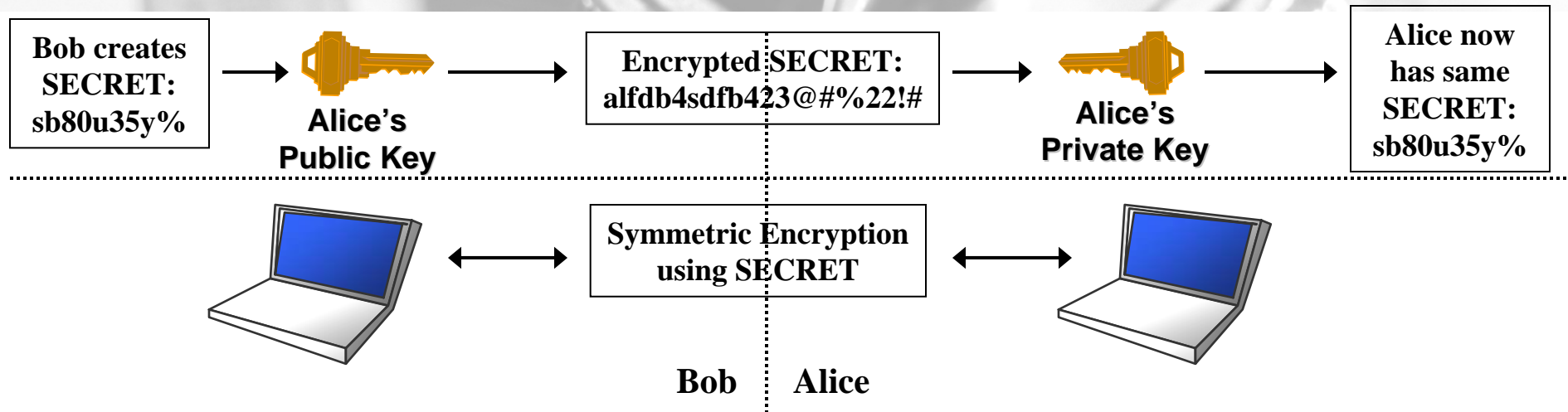
- Two mathematically-related, yet separate keys (based on large prime numbers)
 - private key: secret, not shared, stored encrypted
 - public key: shared, “published” in a public location
- Two main applications:
 - Digital Signatures
 - Symmetric Key Transport

Digital Signatures



- Encrypt a hash of the message (encrypting whole message is more costly, but possible)
- Client receives message, hashes it, decrypts the sender's hash (using sender's public key or their own private key) and verifies

Symmetric Key Transport



- Create a secret key, encrypt it for a specific recipient using the recipient's public key
- Only that recipient can decrypt the message (containing a shared secret) using their private key

What PKI...

- Is:
 - Authentication: who you are
 - Positive identification of other identities
 - Asynchronous in nature
- Is not:
 - Authorization: what you can do (now that we know you)
 - Fast for encryption of large data payloads

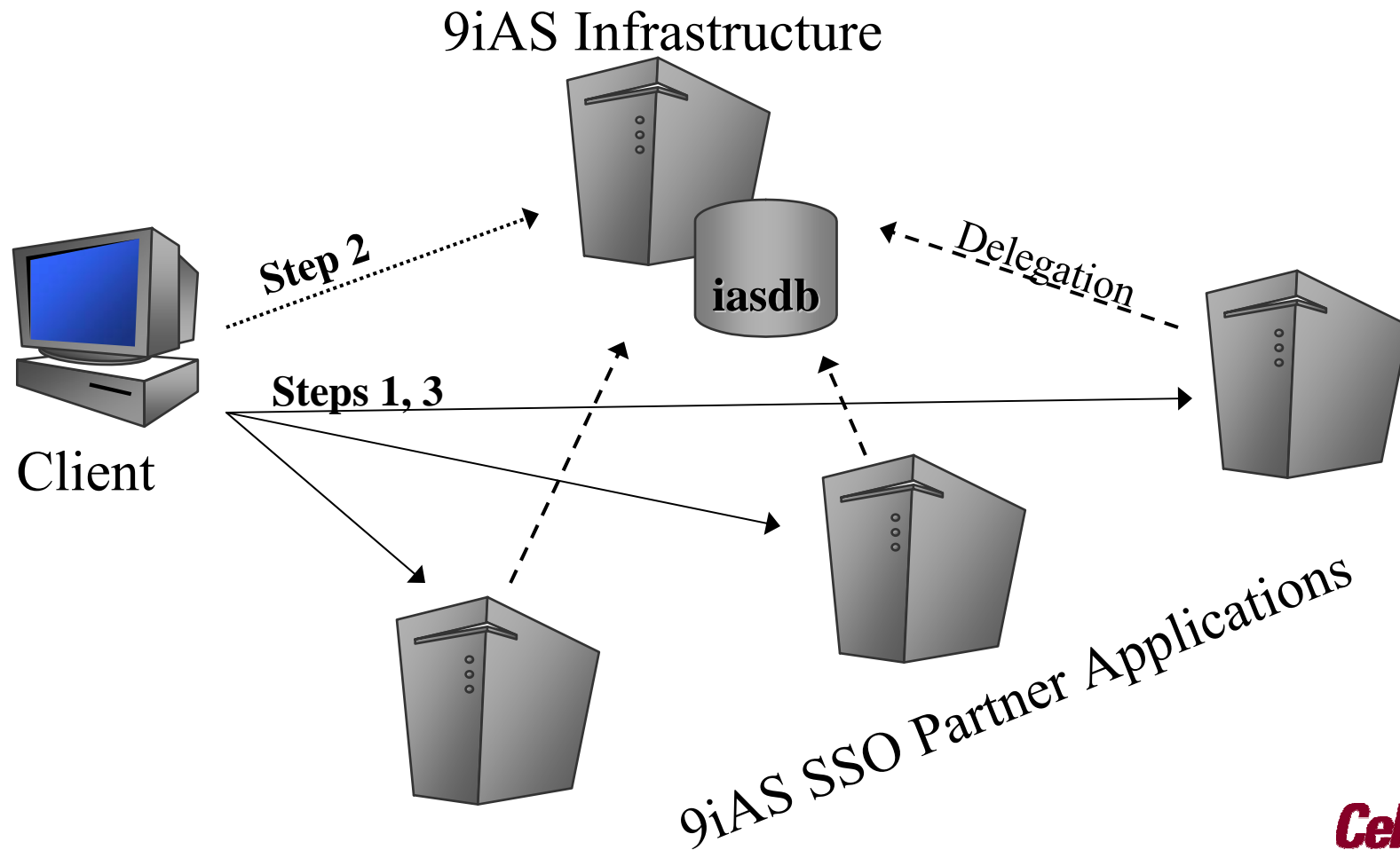
PKI Components

- Certification Authority (CA)
- Registration Authority (RA)
- Online repository (usually LDAP or X.500)
- End entities (users, computers, applications)
- Certificate Revocation List (CRL)
- Public Key Certificate (X.509v3)

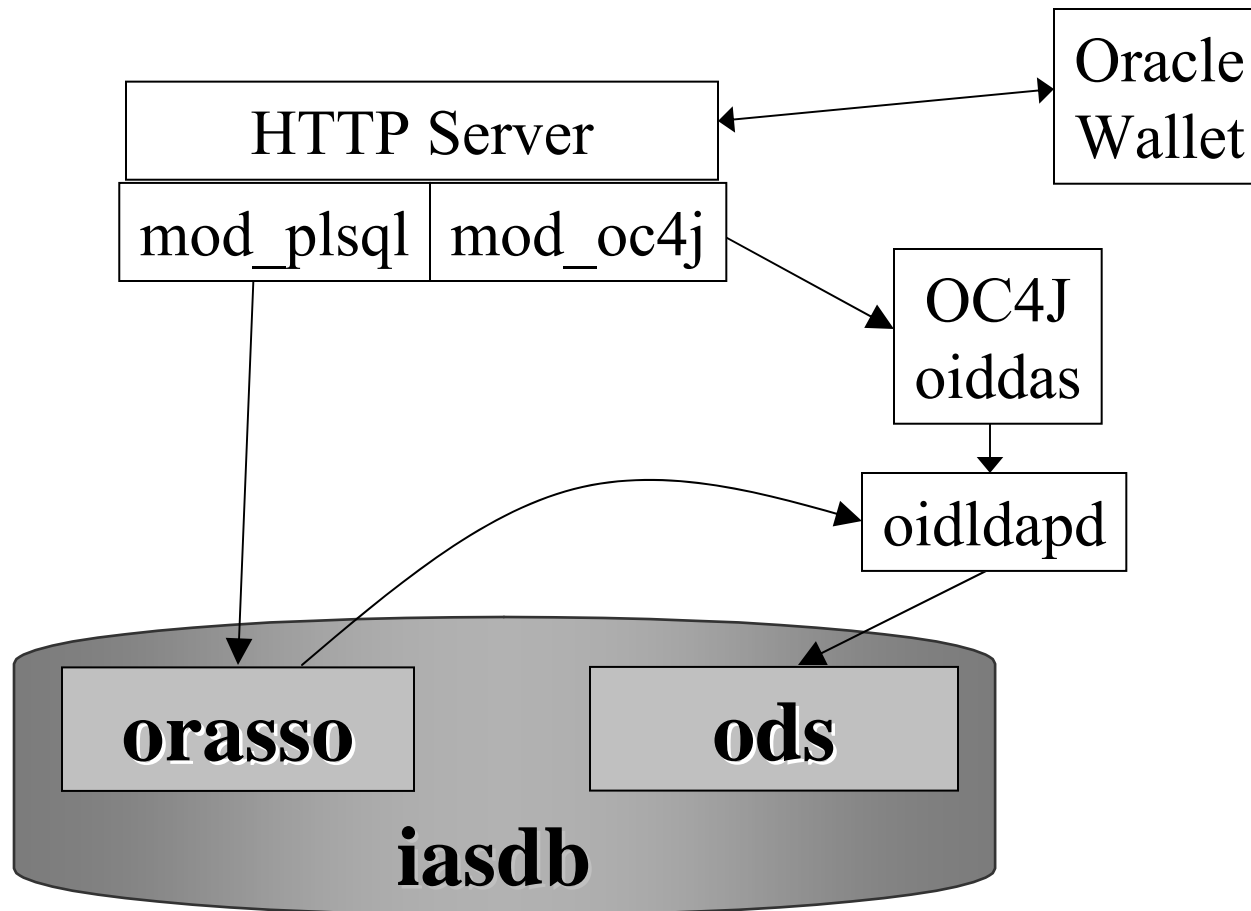
Agenda

- What PKI is and is not
- ▶ 9iAS SSO overview
- 9iAS SSO with digital certificates
- 9iAS SSO third-party integration point
- Web-based PKI sign-on
- Steps to integrate 9iAS SSO with web-based PKI sign-on
- Q & A

9iAS SSO Architecture



9iAS SSO Architecture



9iAS SSO Login Flow (normal)

1. User attempts to access 9iAS SSO partner application (i.e. Portal)
2. Partner app redirects user to SSO server for authentication

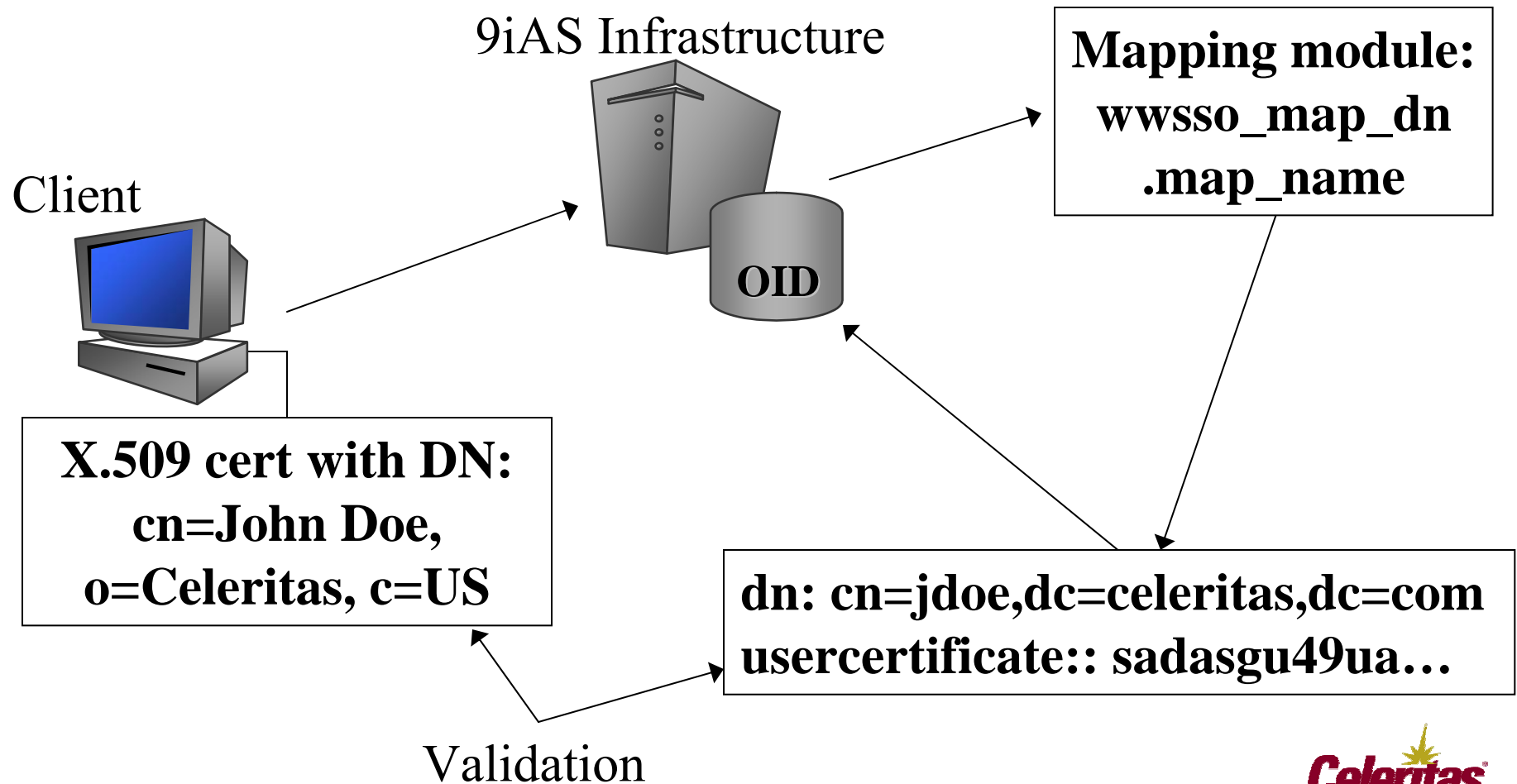
9iAS SSO Login Flow (normal)

3. User submits their username/password via standard login form.
4. SSO Server locates DN for user in OID and attempts to bind to LDAP directory as that DN using the password given
5. If LDAP bind successful, SSO cookies are set for partner application and user is redirected back to partner application.

Agenda

- What PKI is and is not
- 9iAS SSO overview
- ▶ 9iAS SSO with digital certificates
- 9iAS SSO third-party integration point
- Web-based PKI sign-on
- Steps to integrate 9iAS SSO with web-based PKI sign-on
- Q & A

9iAS SSO with digital certificates



9iAS SSO Login Flow (w/ certs)

1. User attempts to access 9iAS SSO partner application (i.e. Portal)
2. Partner app redirects user to SSO server for authentication
3. Browser sends client's X.509 certificate to server.

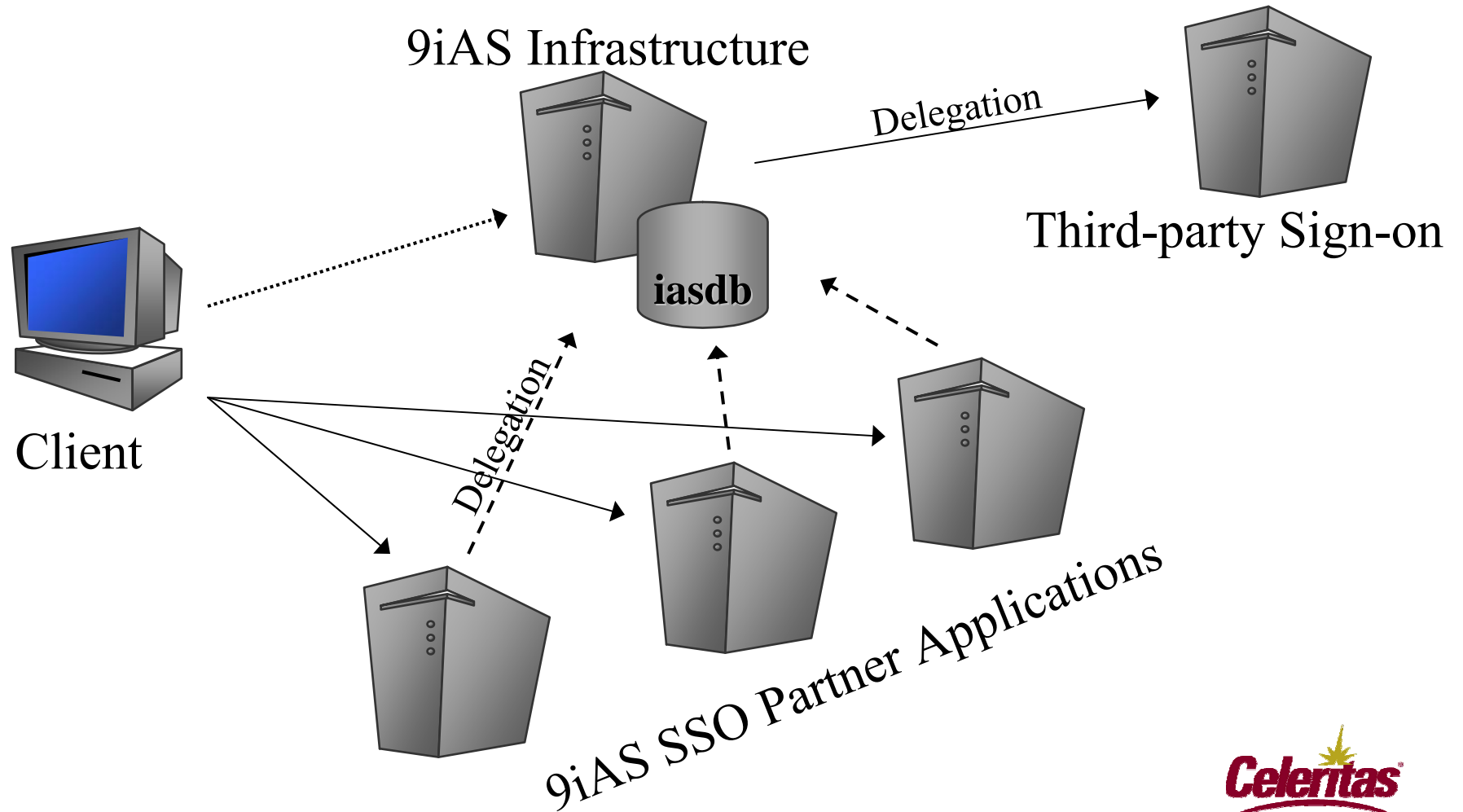
9iAS SSO Login Flow (w/ certs)

4. SSO Server runs DN mapping procedure to map DN from certificate to DN in LDAP
5. Comparison of X.509 certificate with usercertificate attribute of LDAP entry determines whether authentication is successful or not

Agenda

- What PKI is and is not
- 9iAS SSO overview
- 9iAS SSO with digital certificates
- ▶ 9iAS SSO third-party integration point
- Web-based PKI sign-on
- Steps to integrate 9iAS SSO with web-based PKI sign-on
- Q & A

9iAS SSO Third-Party Integration



9iAS SSO Integration Point

- 9iAS 9.0.2 Single Sign-on Administrator's Guide, Chapter 5, details the `WWSSO_AUTH_EXTERNAL` package interface.
- `authenticate_user` function is executed before the default 9iAS SSO login dialog is displayed
- If `authenticate_user` returns a valid SSO username, the SSO server “trusts” this to be the username and the 9iAS SSO cookies are set accordingly.

9iAS SSO Login Flow w/ 3rd Party

1. User attempts to access 9iAS SSO partner application (i.e. Portal)
2. Partner app redirects user to SSO server for authentication
3. In order to access the mod_plsql SSO DAD, the user must first authenticate with the third-party application

9iAS SSO Login Flow w/ 3rd Party

4. Once the third-party application authenticates the user, the original request is submitted to the HTTP server, and `wwsso_auth_external.authenticate_user` is executed.
5. `authenticate_user` reads an HTTP header set by the third-party module to match third-party user to SSO username (may require lookups)

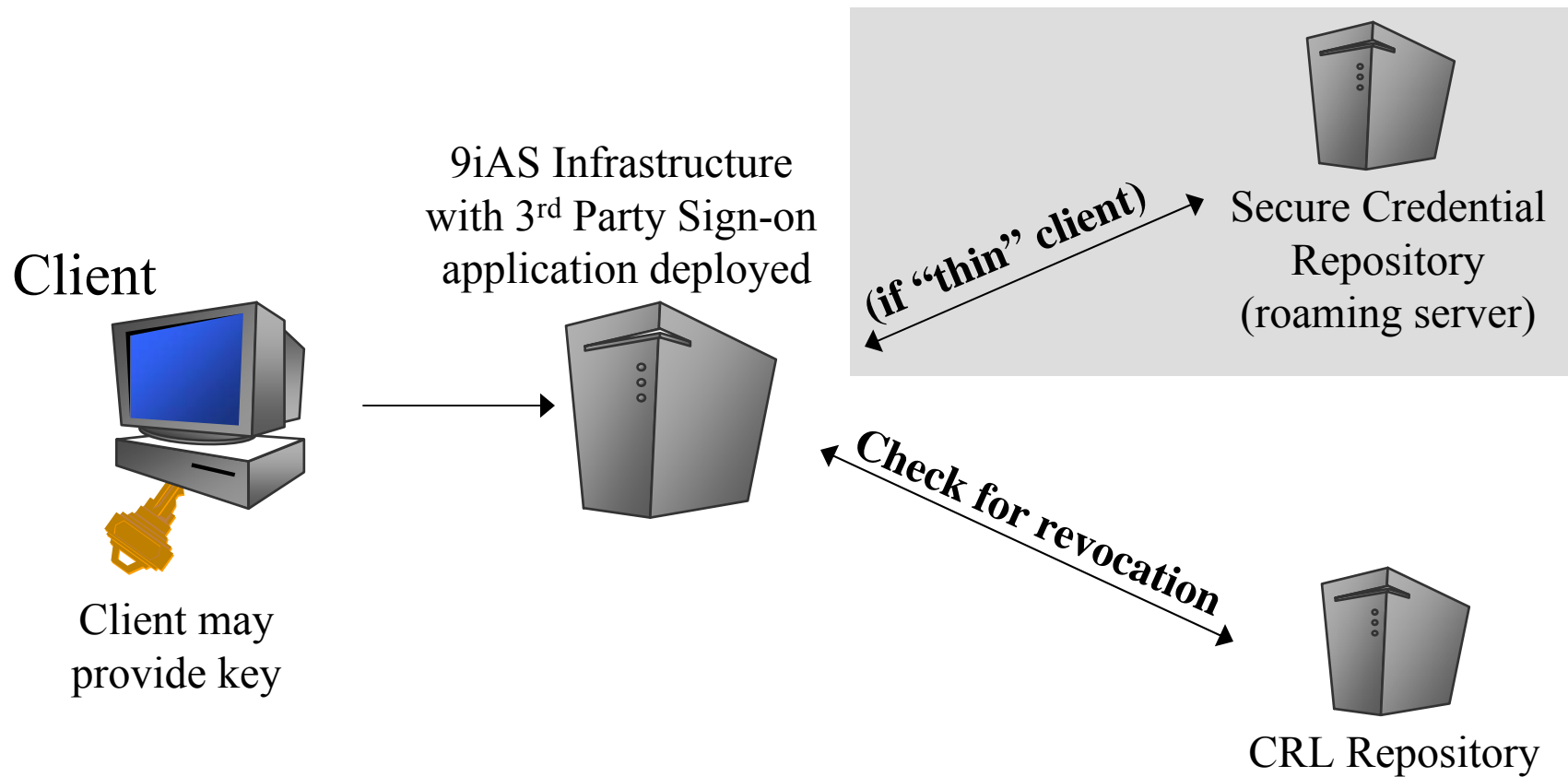
9iAS SSO Login Flow w/ 3rd Party

6. Once SSO username is determined, p_user OUT parameter is set and authenticate_user procedure exits
7. SSO server identifies user by setting 9iAS SSO cookie for partner application and redirects user back to originally-requested partner application

Agenda

- What PKI is and is not
- 9iAS SSO overview
- 9iAS SSO with digital certificates
- 9iAS SSO third-party integration point
- ▶ Web-based PKI sign-on
- Steps to integrate 9iAS SSO with web-based PKI sign-on
- Q & A

Web-based PKI Sign-on



Agenda

- What PKI is and is not
- 9iAS SSO overview
- 9iAS SSO with digital certificates
- 9iAS SSO third-party integration point
- Web-based PKI sign-on
- ▶ Steps to integrate 9iAS SSO with web-based PKI sign-on
- Q & A

Integration Steps for SSO w/PKI

1. Deploy 3rd-Party Sign-on with 9iAS Infrastructure
2. Test 3rd-Party Sign-on independently (no SSO integration yet)
3. Configure 9iAS Infrastructure HTTP server to for 3rd-party integration

Integration Steps for SSO w/PKI

4. Create and install
WWSSO_AUTH_EXTERNAL package
body
5. Protect the orasso mod_plsql DAD with
3rd-party protection
6. Test/Debug

Actual Implementation Sign-on Flow

1. User accesses partner application and is redirected to 9iAS SSO in 9iAS Infrastructure (/pls/orasso?...)
2. HTTP server in 9iAS Infrastructure is configured with 3rd-party module to protect /pls/orasso with 3rd-party sign-on
3. 3rd-party sign-on module takes over and presents sign-on dialog, authenticates user

Actual Implementation Sign-on Flow

4. User is allowed to access /pls/orasso?... (original request) and `authenticate_user` function is run to read `HTTP_ENTRUST_CLIENT` header to determine user's DN
5. DN is used to look up user in OID. If exists, set username to RDN of entry found and return. Else, look up user information in Entrust directory and create new OID entry, then return SSO username

Actual Implementation Sign-on Flow

6. SSO receives authoritative username from authenticate_user function and sets appropriate SSO cookies
7. User is redirected back to partner application with proper 9iAS SSO cookies in place to identify them

Troubleshooting Tips

- NOTE 198732.1 (debug_print & wwssso_log\$)
- Use UTL_FILE in your custom code (wwssso_auth_external) for debugging and audit trail
- DBMS_LDAP does not seem to work with all LDAP directories
- **LogLevel debug** in httpd.conf exposes tons of information

IOUG RAC SIG Events

- Today, 12 noon, Room 709: Expert presentation “Workload Distribution in a RAC Environment”
- Tomorrow (Tuesday), 12 noon, Room 709: RAC SIG Roundtable—”Stump Your Peers”
- Lunch provided both days at Room 709



Q&A



Survey me please!

- Dan Norris
- Session 100: Single Sign-on Integration with PKI
- Monday, April 19th, 10am
- Questions later? Send me an email at norris@celeritas.com.