SSDS: Secure Session-Data Storage

Protecting HTTP session-data on web application servers from prying eyes

Introduction

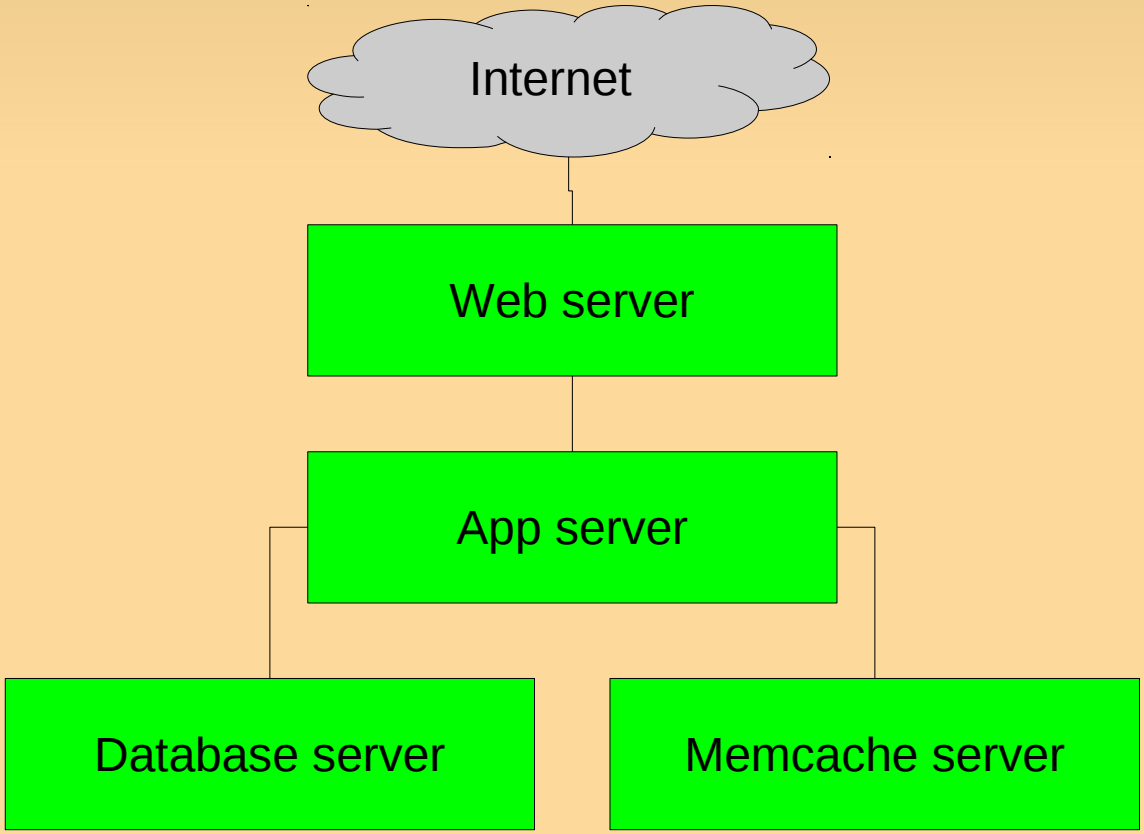
- Professional life
  - Information Security Officer at Deutsche Post
  - http://blog.deutschepost.de/security/ (soon)
- Personal life
  - http://juergen.pabel.net/blog/
  - http://www.rugby-koeln.de/

Agenda

- Standard web application architecture
  - HTTP session-data basics
  - Associated risks
- Secure Session-Data Storage (SSDS)
  - Concept
  - Cryptographic details
- Demo (php-ssds)

Context

- The web application has been audited for vulnerabilities (Cross-Site-Scripting, SQL Injections, …)
- The servers and networks have been hardened



Standard web application architecture

- Cookie based state & session-management
  - Random ID assigned to each client as cookie value

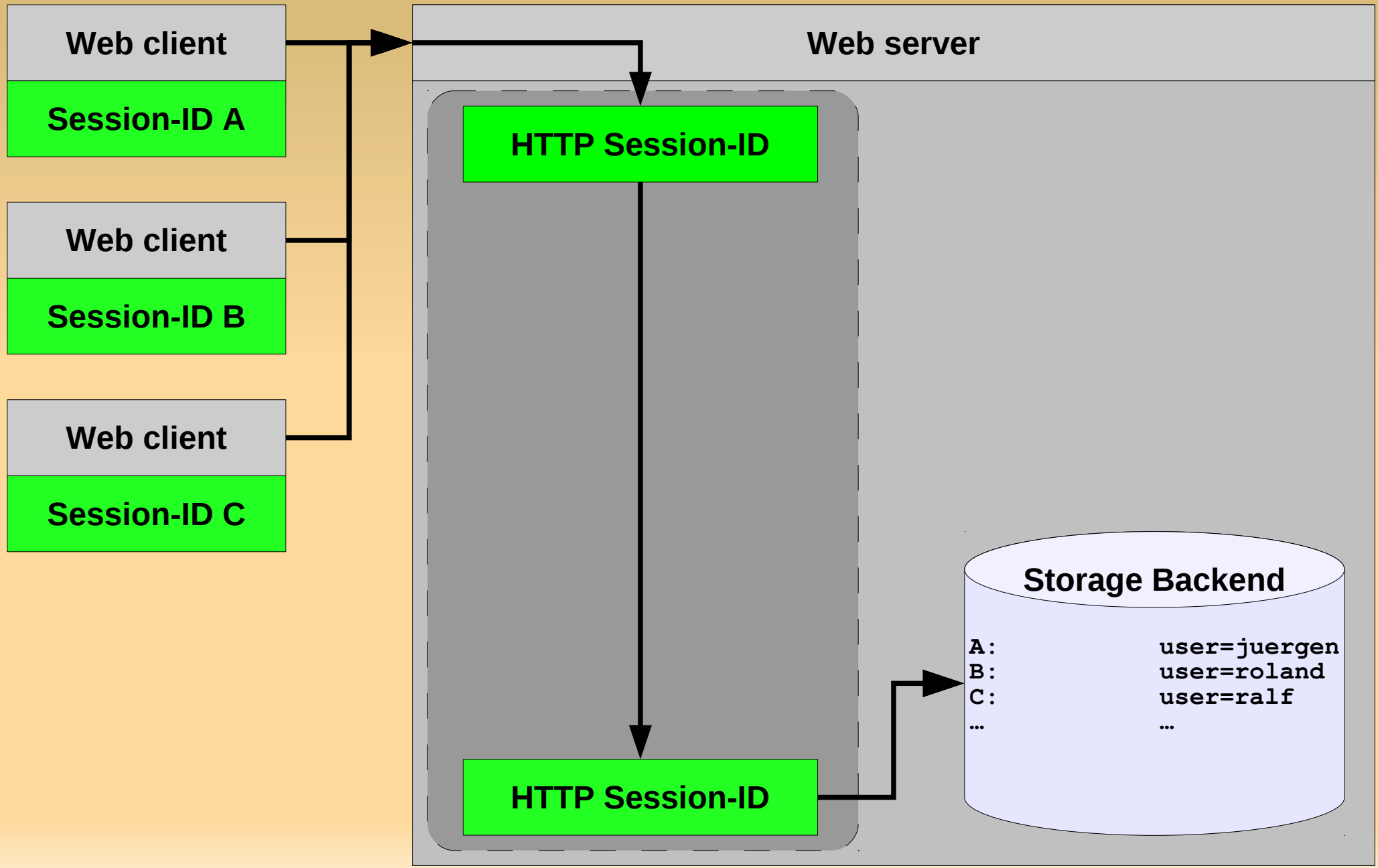
```
HTTP/1.1 200 OK
Server: example.com
Set-Cookie: PHPSESSID=A1B2C3D4E5F6G7H8
[...]
```

  - Server maintains ”database” of all IDs and their associated data values
  - Client sends cookie value with each HTTP request

```
GET / HTTP/1.1
Host: example.com
Cookie: PHPSESSID=A1B2C3D4E5F6G7H8
[…]
```

- Permanent vs. Session cookies

Cookies in web applications



Associated risks (1/2)

- Session-ID hijacking
  - The Session-ID is used as the primary key in the session-data storage backend (filesystem, memcache,...)
  - Access to the session-data storage backend yields access to any currently logged-in user session
    - Copy any Session-ID from the storage backend and put it in the cookie in a browser

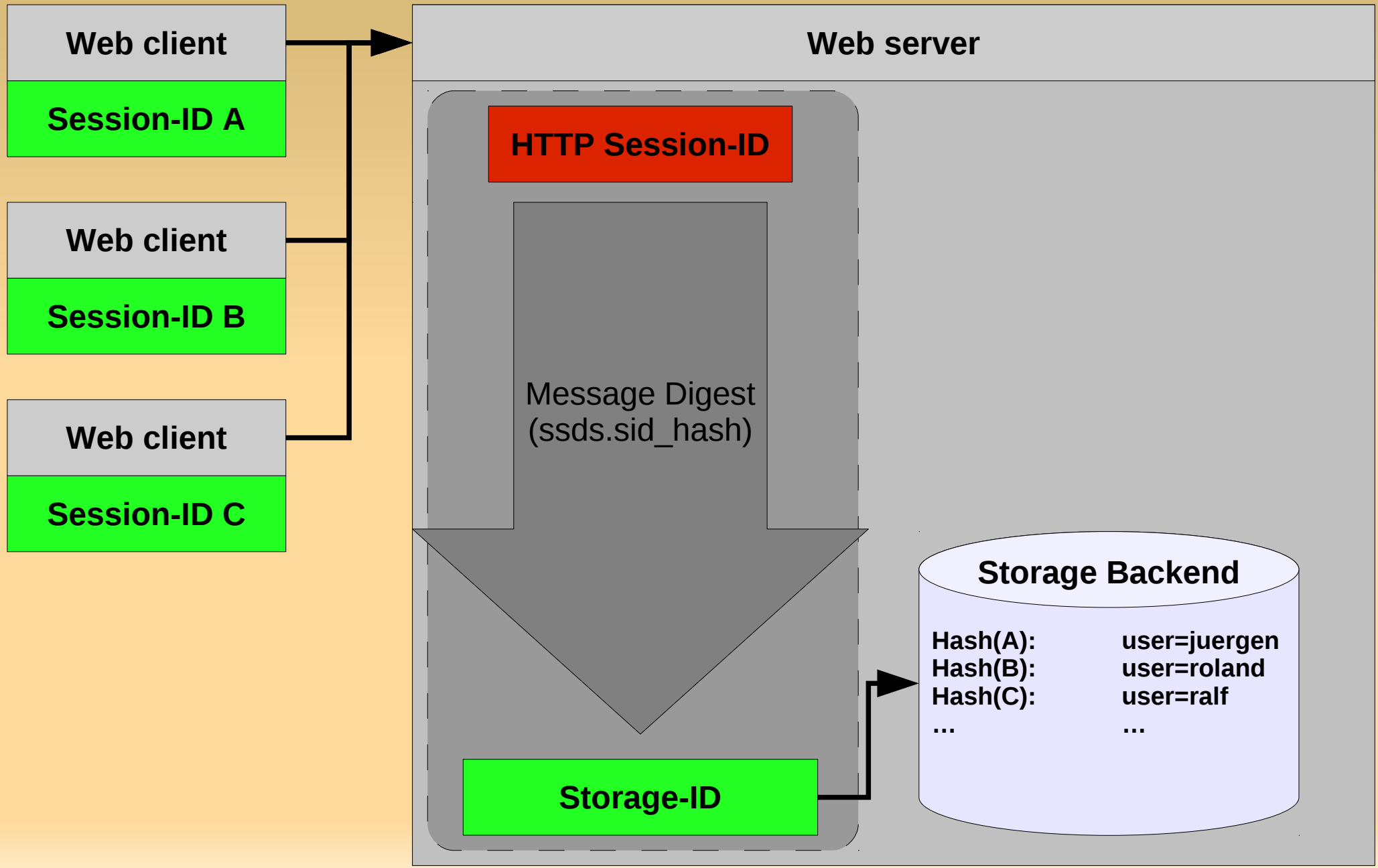
Associated risks (2/2)

- Name five things that....are usually/sometimes ”stored” on the server with a cookie
  - HTTP state data (login status, multi-page status, ...)
  - Account data (username, privileges, ...)
  - Application data (shopping cart, todo list, ...)
  - Passwords (user login, backend/partner system, ...)
  - Credit cards

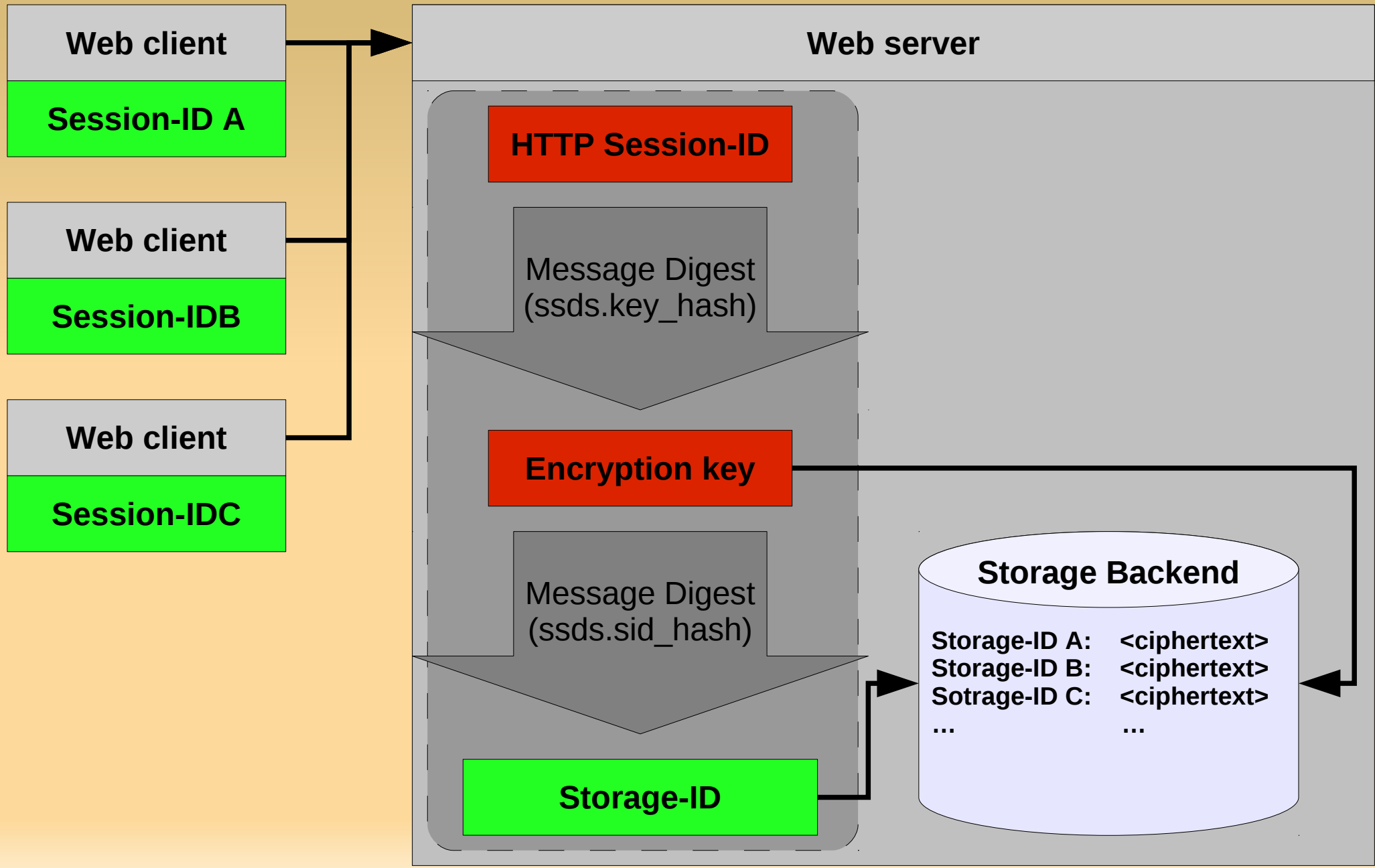
The concept of SSDS

- Hash the Session-ID before passing it to the storage backend as the primary key (”Storage-ID”)
- (Optionally) Encrypt the session-data using the Session-ID...
  - ...however, Session-IDs might be of variable length...
  - ...so the (fixed length) hash-value of a Session-ID is used as the encryption key...
  - ...but that would be the Storage-ID; thus with session-data encryption enabled, the Storage-ID is actually a hash-value of the encryption key.

SSDS concept visualized: Session-ID



SSDS concept visualized: Encryption



SSDS cryptographic details

- Encryption key remains constant (for any given Session-ID) but multiple encryptions occur
  - A unique initialization-vector (IV) needs to be provided for each encryption operation...
  - ...and for the most common block mode cipher (CBC), the IV also needs to be unpredictable...
  - ...and the first idea was to use the server's PRNG but that might drain the entropy pool very quickly
    - ...thus, in SSDS the IV is computed as $\mathtt{hash}_{\mathtt{iv\_hash}}(\mathtt{concat(NOW,SESSION\text{-}ID)})$

php-ssds

- Implemented as a PHP extension in C
- Registers a PHP ”save_handler”
  - Uses a backend/pass-through ”save_handler” for actually persisting all data (like ”files”, ”mysql”, etc)
- Configuration per PHP settings
  - `ssds.save_handler = <backend for data storage>`
  - `ssds.save_path = <configuration for storage backend>`
  - `ssds.sid_hash = <digest for Storage-ID derivation>`
  - `ssds.data_cipher = <cipher for data encryption>`
  - `ssds.key_hash = <digest for key derivation>`
  - `ssds.iv_hash = <digest for IV derivation>`

php-ssds Configuration

- php.ini

```
session.save_handler = ssds
session.save_path =
```

- ssds.ini

```
extension = ssds.so
ssds.save_handler = files
ssds.save_path = /var/lib/php5
ssds.sid_hash = ripemd160
ssds.data_cipher = aes-256-cbc
ssds.key_hash = sha256
ssds.iv_hash = md5
```

php-ssds Demo?

- Nothing to see here, move along!
  - Really; it runs entirely transparently for all web applications
- Example for Session-ID **viq6ehuba8lb9gpg6g1hi7g3n7**
  - php (/var/lib/php5/sess_**viq6ehuba8lb9gpg6g1hi7g3n7**)
    `time|i:1337337184;data|s:1:"x";`
  - php-ssds (/var/lib/php5/sess_**e73407a8d1ac72ca03c599b660ae9ae7**)
    `9Ar5gGImihQ7B+/g7m6l+4it9VdU8/Y/b2c5LFk2IPY=#9281b50004c880e0e317615c9d7fd2fa`

*-ssds

- php-ssds
  - Version 1.1
  - Stable, tested and security audited
- java-ssds
  - Work in progress
- {python|ruby|.NET|*}-ssds
  - Please contact me if you want to work on this

Pointers

- Sources: http://php-ssds.sourceforge.net/
  http://java-ssds.sourceforge.net/ (~Nov)
- Blogs: http://juergen.pabel.net/blog/
  http://blog.deutschepost.de/security/
- Contact: juergen@pabel.net
  juergen.pabel@deutschepost.de

Q & A

Please ask questions!