CAS を利用した Single Sign On 環境の構築

内藤 久資

naito@math.nagoya-u.ac.jp

名古屋大学多元数理科学研究科

Plan of Talk

- CAS および CAS^2 の簡単な紹介
- CAS を利用した Single Sign On 環境の実例の紹介
  名古屋大学ポータル
- CAS 及び CAS^2 の仕組み
- CAS の利点

CAS & CAS^2 とは

- CAS ： Yale 大学が開発した Open Source software
  - Web Application のための Authentication 環境
  - 現在は JA-SIG の Official Project
  - 強力な “Authorization” 機能を追加 (CAS^2)
- 特徴
  - Web Application に対する認証に特化したシステム
  - Single Sign On 環境を容易に実現可能
  - Web Application 側には特権が必要ない

デモ

- 名古屋大学ポータルでの CAS^2 の利用
  - `https://mynu.jp/` 「名古屋大学ポータル」
  - `http://tomcat.math.nagoya-u.ac.jp/test/` 「テスト用のサーブレット」
- これら 2 つの Web Application が Single Sign On で利用可能.
- 名古屋大学「新教務システム」などでも利用している

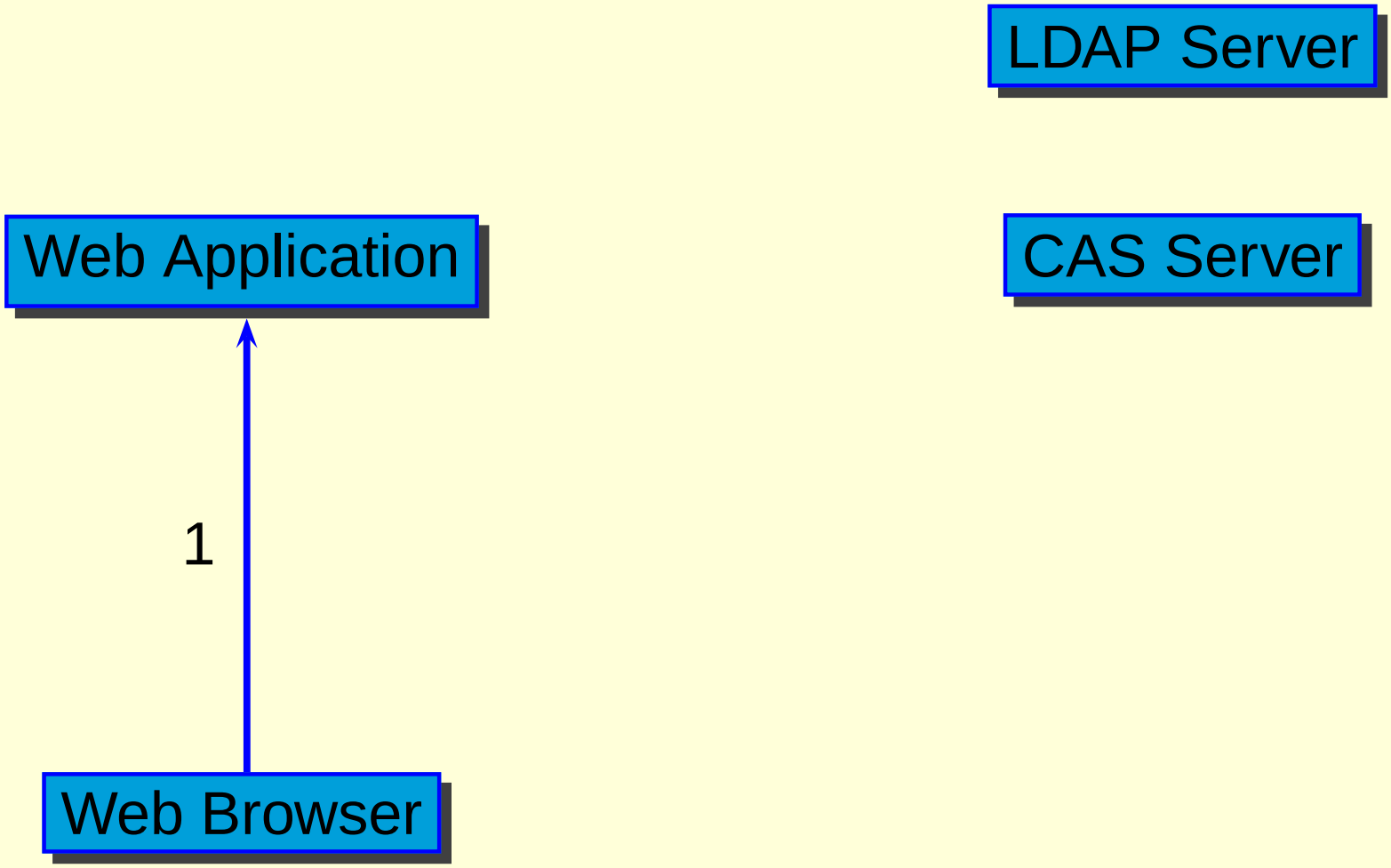
CAS (CAS^2) 認証のしくみ

- 用意すべきもの
  - Web Application Server (including CAS client)
  - CAS Server (over Tomcat)
  - Directory Server (example LDAP Server)
  - Web Browser
- 登場する概念
  - Ticket Granting Cookie (TGC)
  - Service Ticket (ST)

CAS (CAS^2) 認証のしくみ

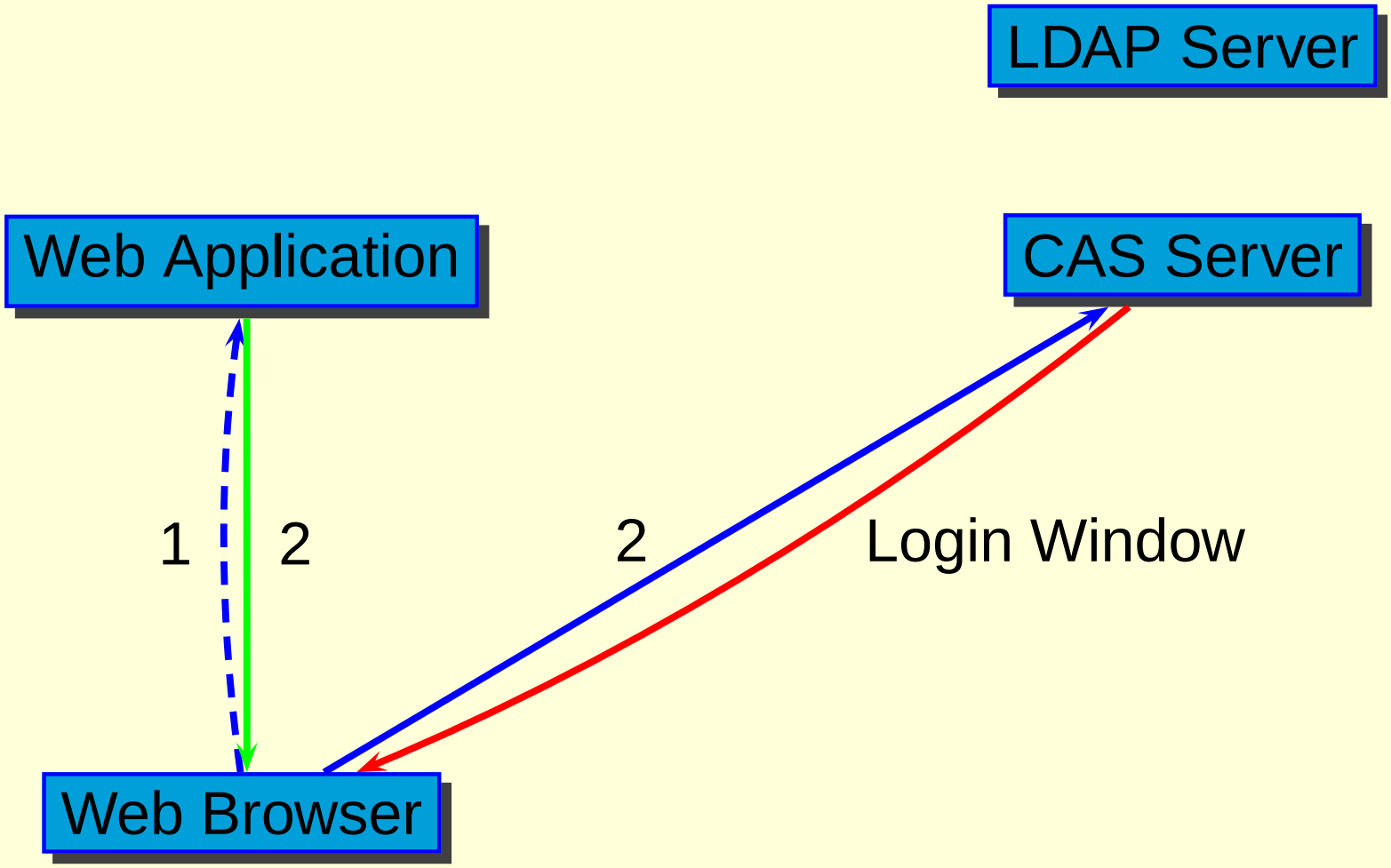
- Login しているユーザのブラウザには TGC が保存される
  - CAS Server は TGC & ST データベースを保存
- １回のアクセスごとに ST を発行
  - ST は One Time Ticket
  - ST は TGC に付随
- TGC で Authentication, ST で Authorization を行う
- ST Varidation → Application に「ユーザのデータ」を送信
- TGC の Timeout ＝ Session Timeout
- TGC の削除 ＝ Logout

CAS 認証のしくみ (1: Login (1))



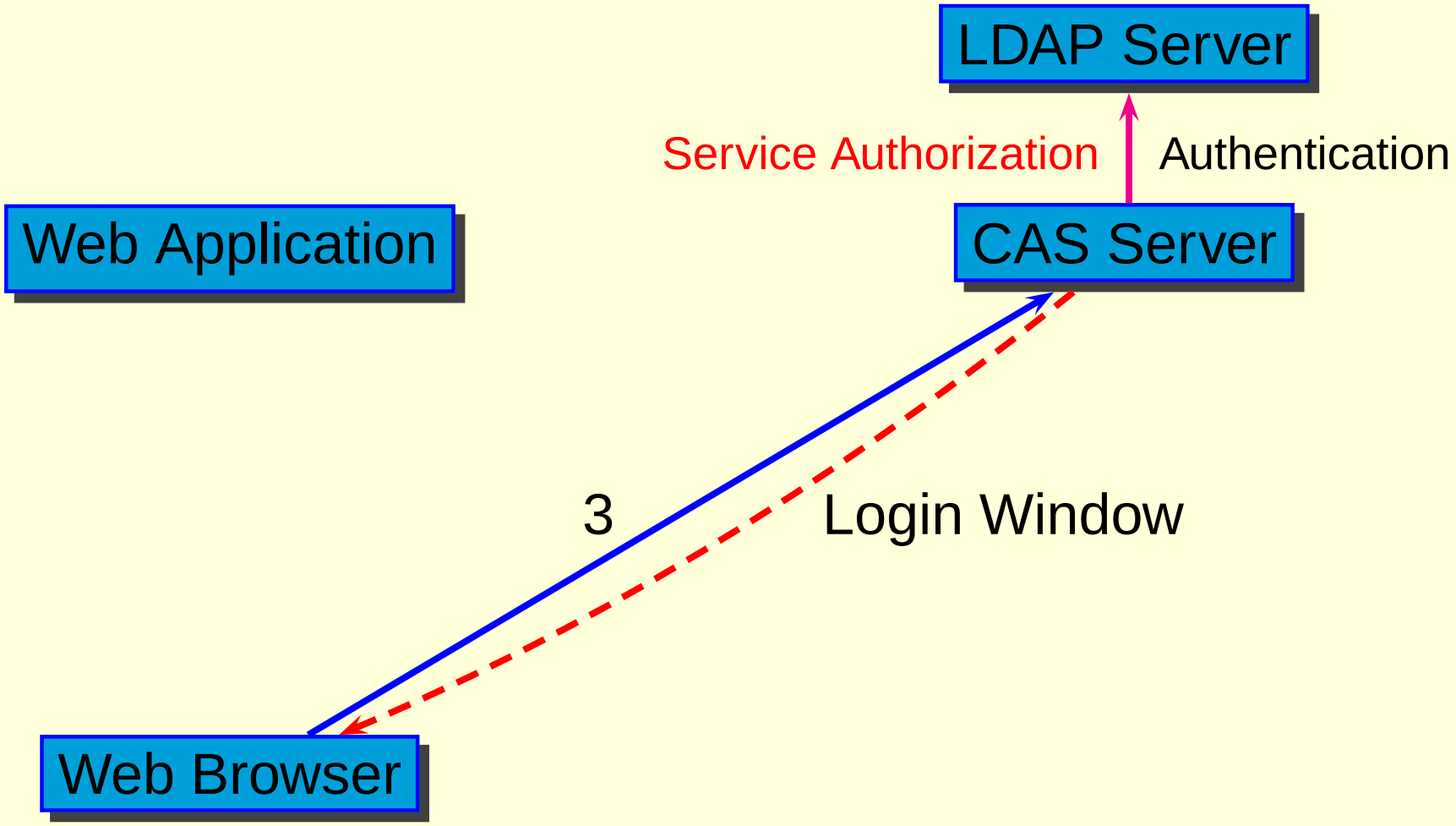
1. Access to **`https://aFQDN/a.html`**

CAS 認証のしくみ (1: Login (2))



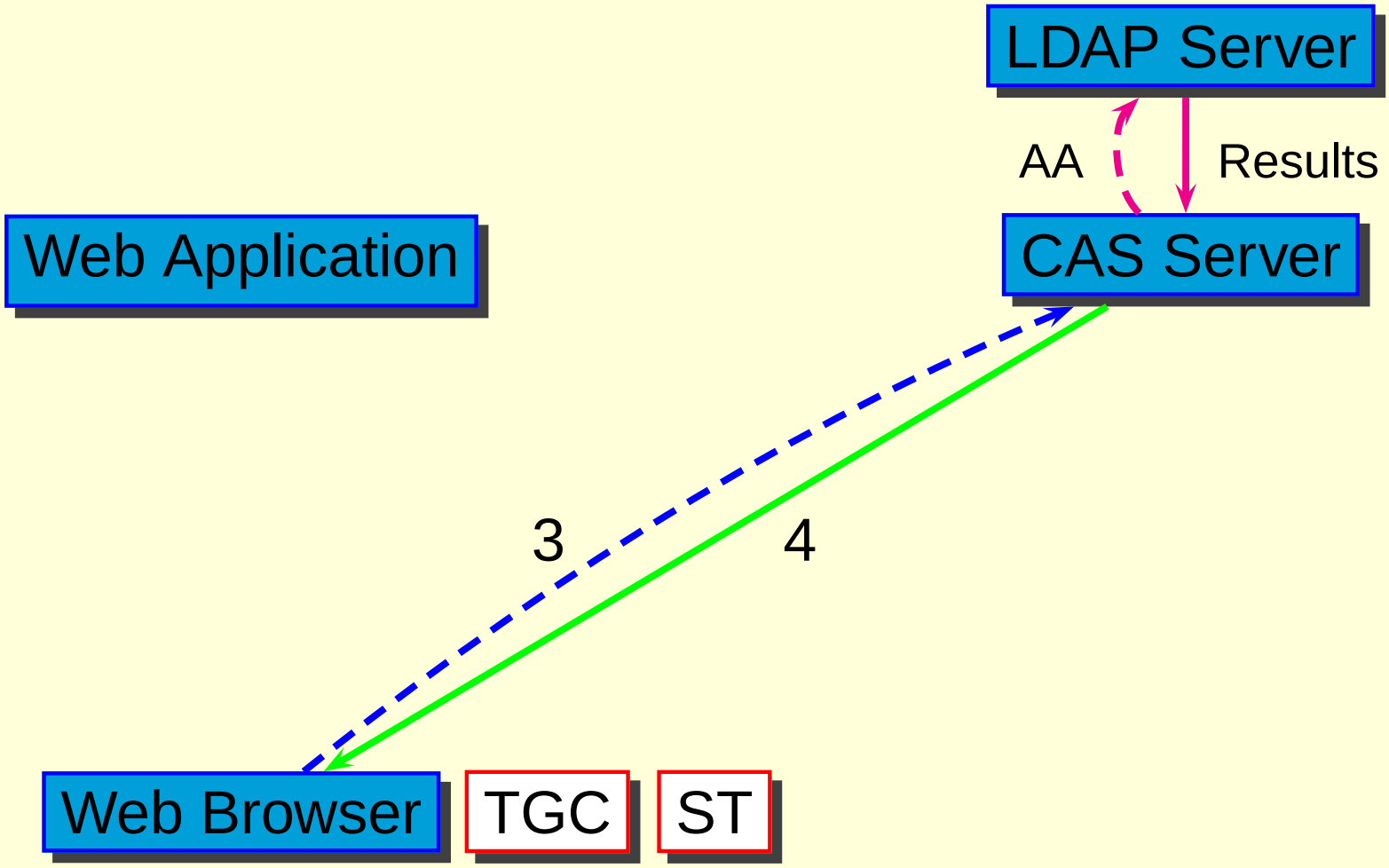
2. Redirect to
   `https://CAS/login&service=https://aFQDN/a.html`

CAS 認証のしくみ (1: Login (3))



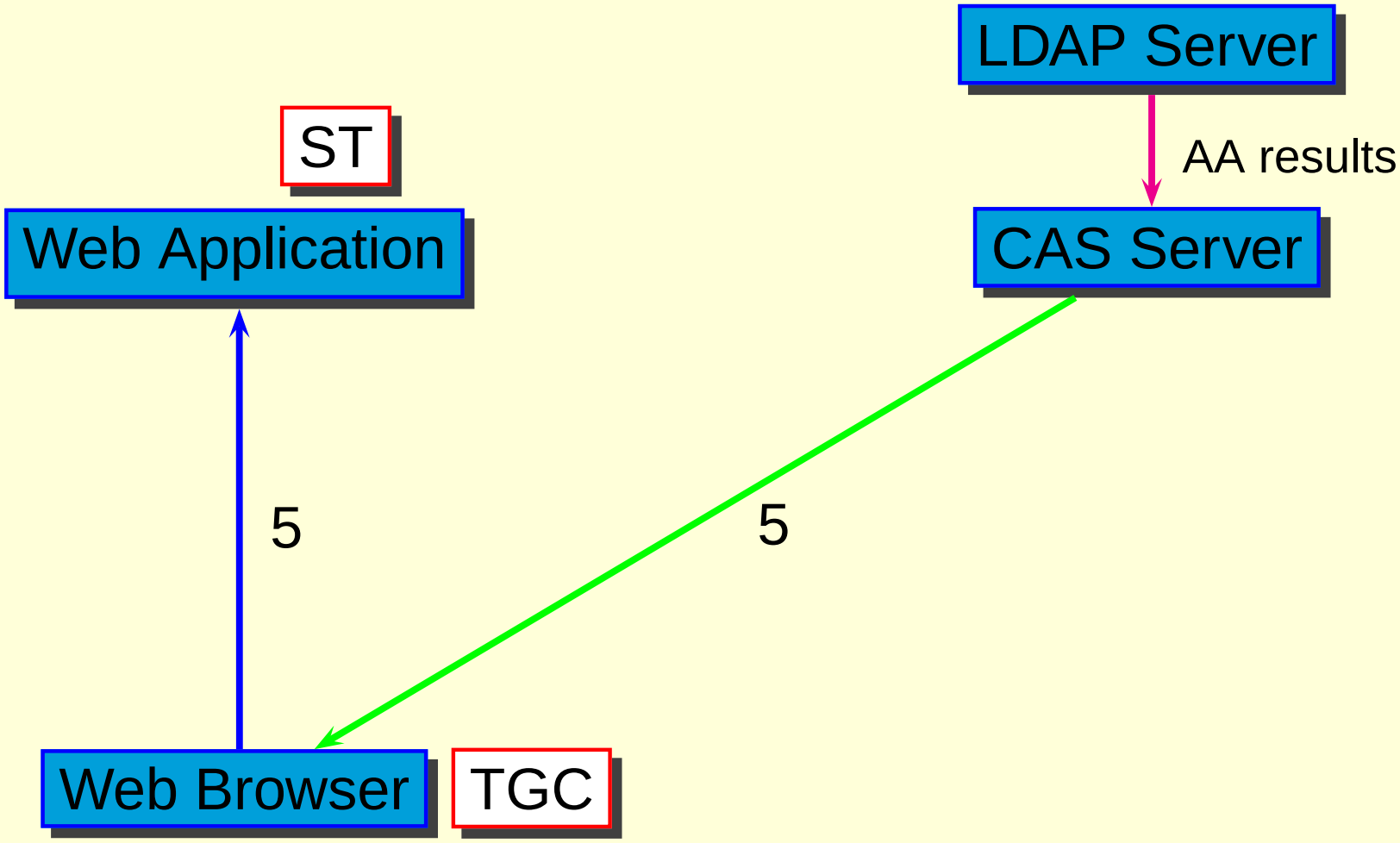
3. Input UserID & Password with service **`https://aFQDN/a.html`**

CAS 認証のしくみ (1: Login (4))



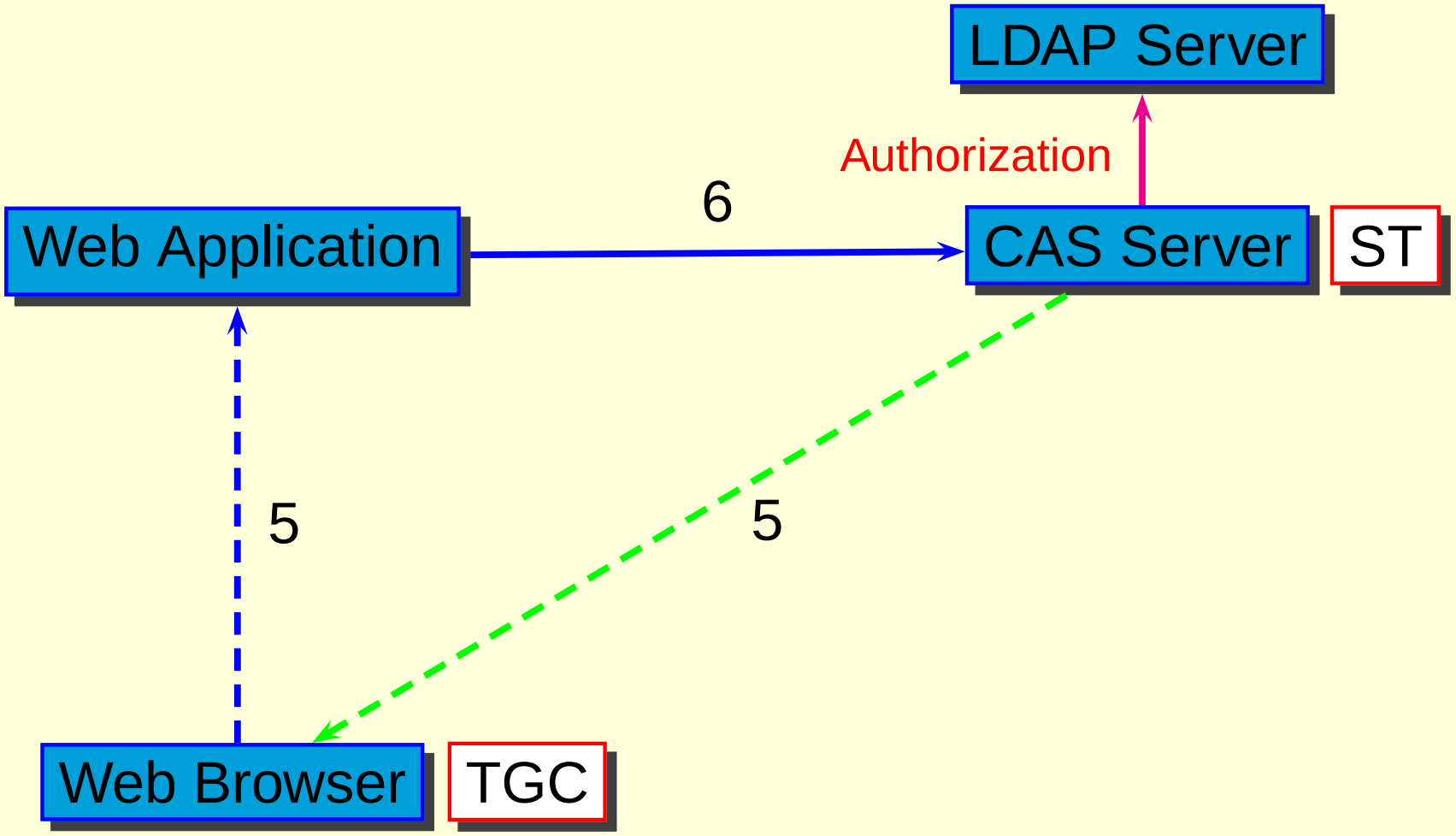
4. Send Ticket Granting Cookie to Browser

CAS 認証のしくみ (1: Login (5))



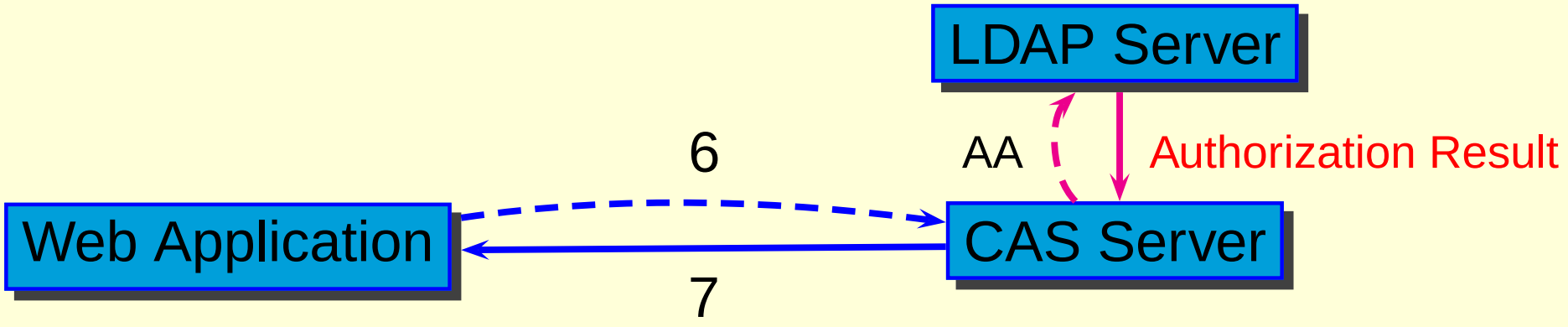
5. Redirect to **`https://aFQDN/a.html&ticket=ST-xxx`**

CAS 認証のしくみ (1: Login (6))



6. Verify Service Ticket

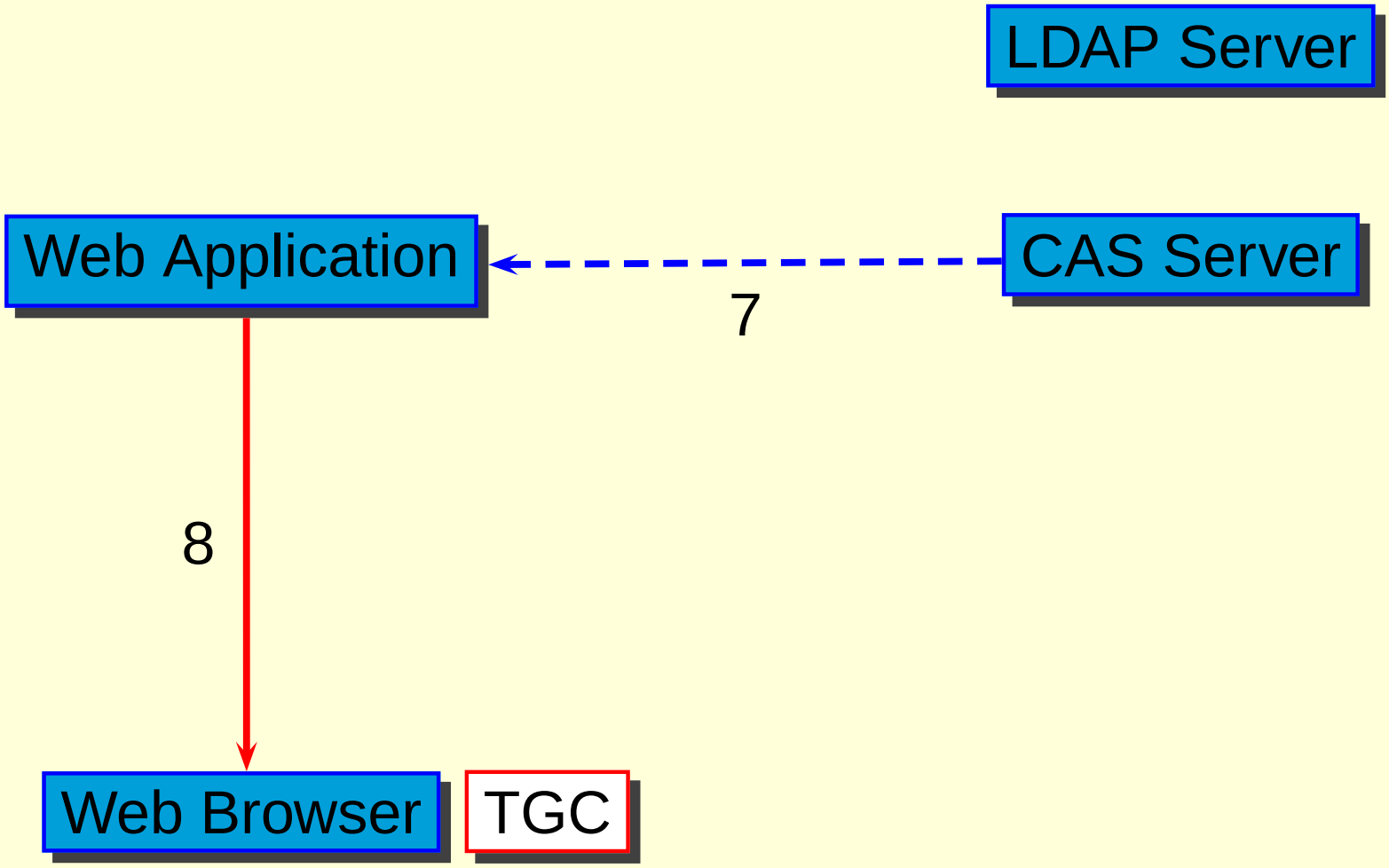
CAS 認証のしくみ (1: Login (7))



Web Browser TGC

7. Receive verify result form CAS server

CAS 認証のしくみ (1: Login (8))



8. Receive Data from Application Server

CAS 認証のしくみ (注意事項)

- Login の一連の操作
  - 一見すると何度もアクセスが発生している
  - JavaScript/HTTP redirection が多数を占める
- 実際にユーザから visible なアクセス：以下の 2 回
  - Login Window
  - 実際のページの取得

CAS 認証のしくみ (2: Verify Ticket)

- Login 後のアクセス
  - ST による Authorization
  - 異なる “Service Class” へのアクセス時には TGC を検証
  - アクセスごとに TGC の “count down timer” を更新
- ST が Timeout している時
  - Login への redirection を発行
  - Authorization 後に新規 ST を発行

CAS 認証のしくみ (2: Verify Ticket (0))

LDAP Server

Web Application

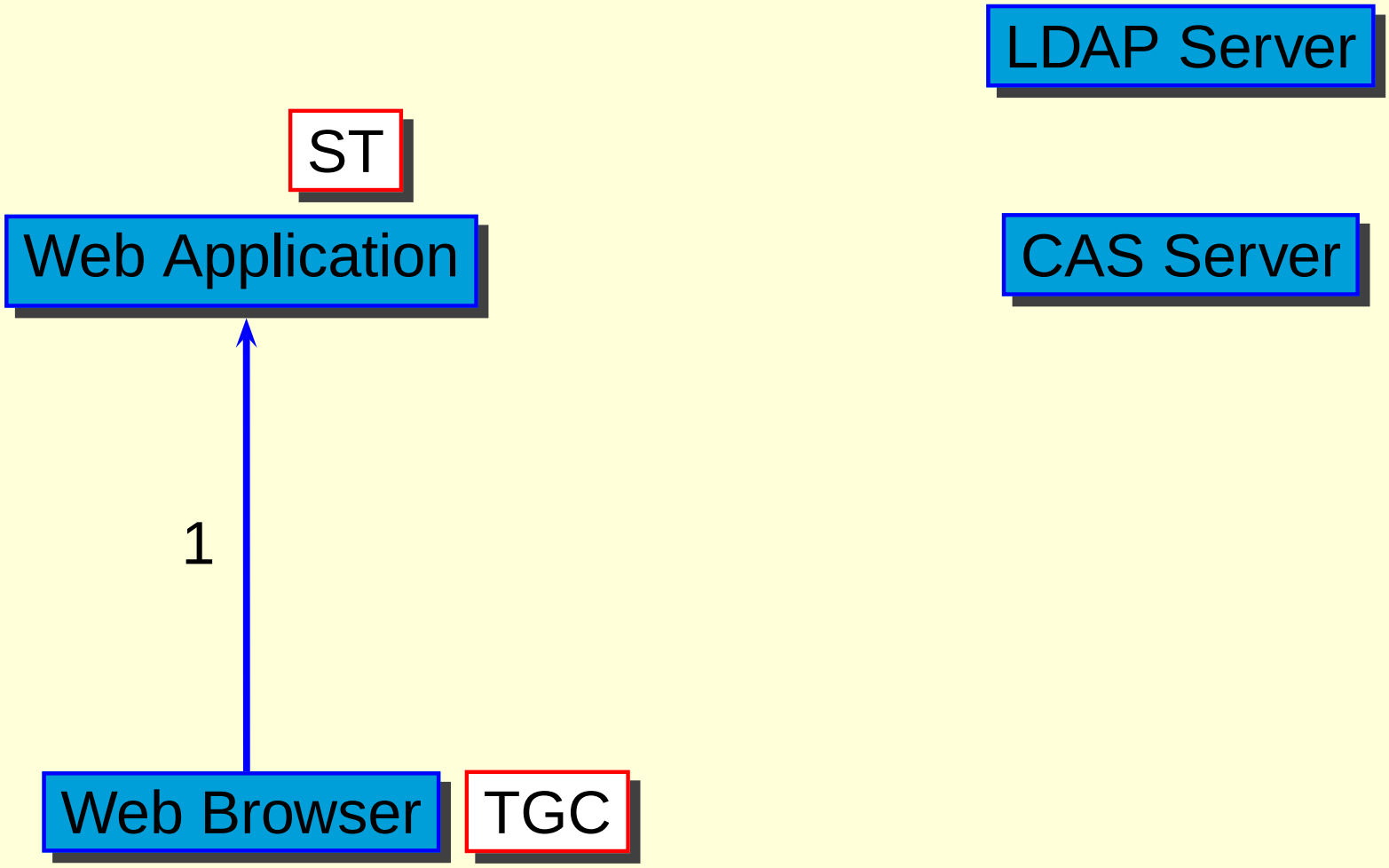
CAS Server

Web Browser

TGC

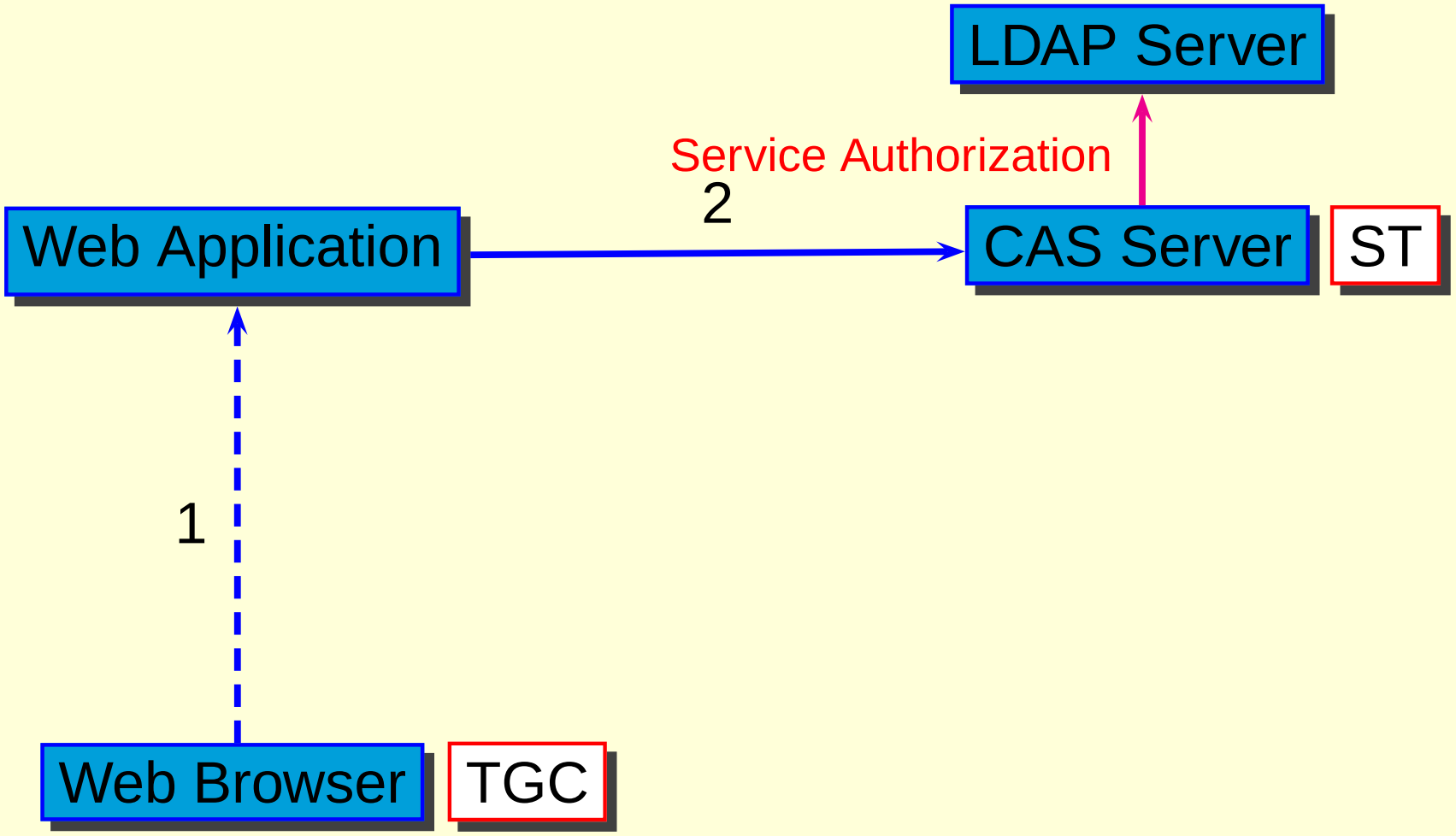
ST

CAS 認証のしくみ (2: Verify Ticket (1))



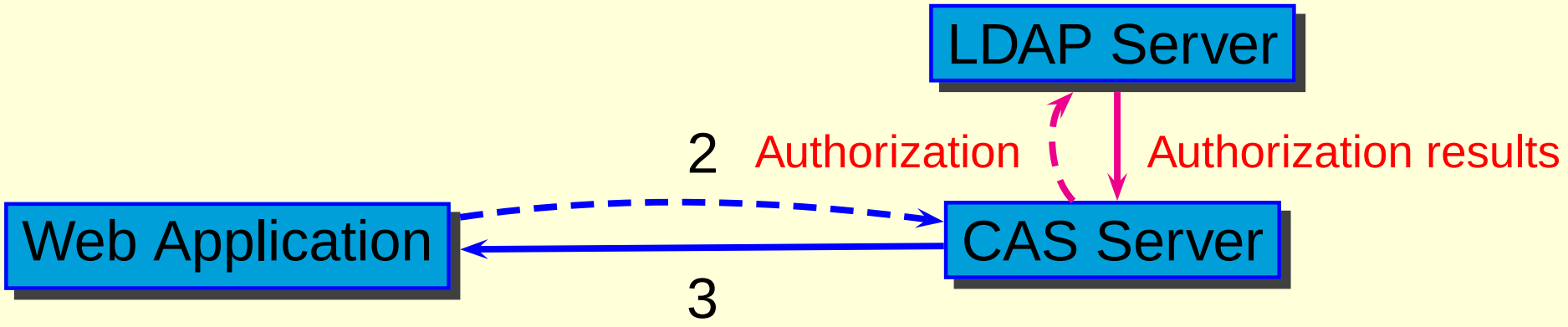
1. Access to **`https://aFQDN/a.html&ticket=ST-xxxxx`**

CAS 認証のしくみ (2: Verify Ticket (2))



2. Verify **`ticket=ST-xxxxx`** with **`service=https://aFQDN/a.html`**

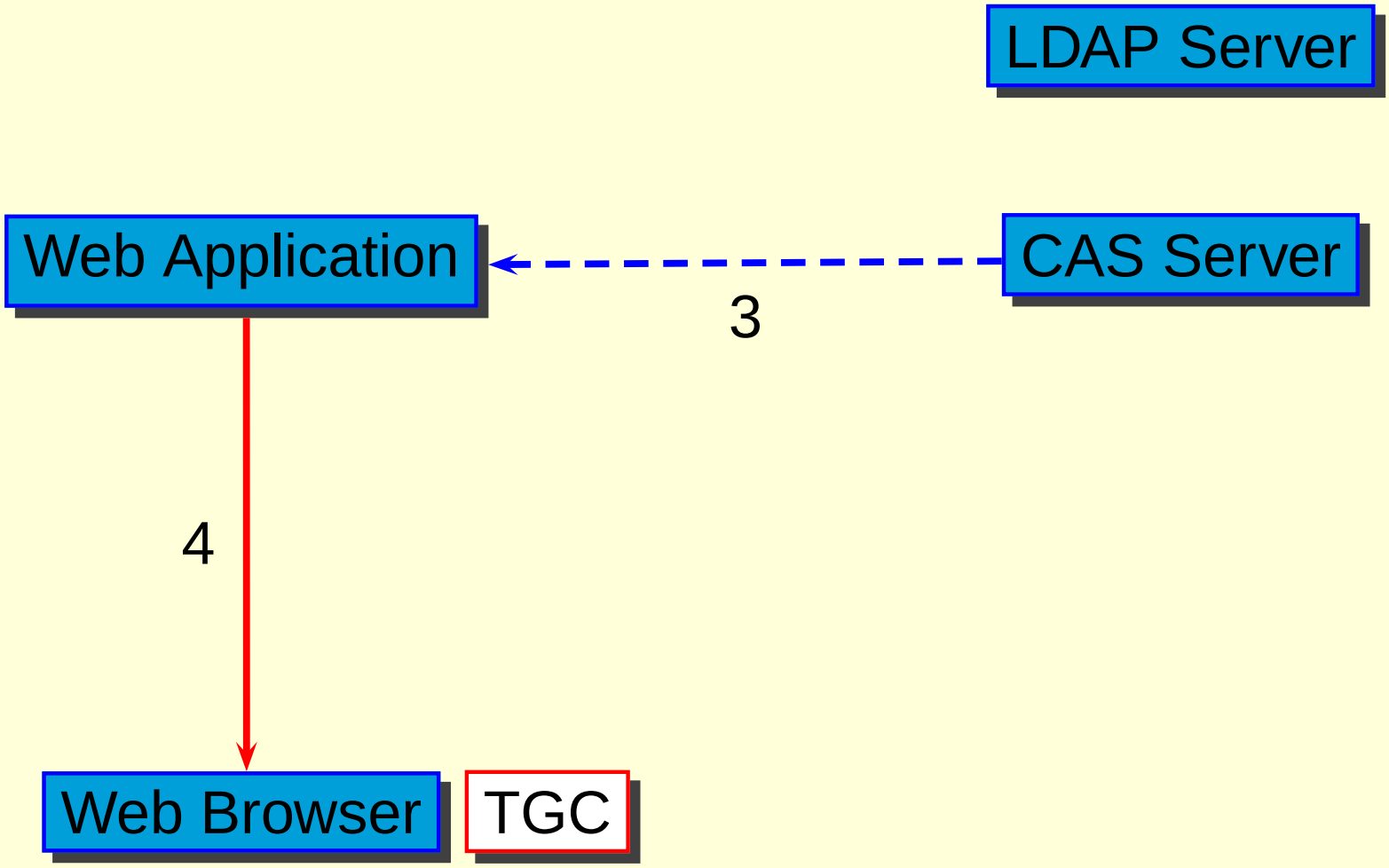
CAS 認証のしくみ (2: Verify Ticket (3))



Web Browser TGC

3. Get authorization results and user infomation

CAS 認証のしくみ (2: Verify Ticket (4))



4. Reply from Web Application

Verify Ticket の注意事項

- Verify Ticket の結果は CAS client が処理する
- CAS client が返すもの (Original の CAS)
  - Ticket Validation の結果
  - ユーザ ID
- CAS client が返すもの (CAS^2)
  - Ticket Validation の結果
  - ユーザデーターベース内の任意の属性値
    - どの属性値を返すかは Application ごとに指定可能

      より正確には「アクセスコントロールクラス」ごとに指定可能
    - CAS client からの戻り値から属性値を取得できる

CAS 認証のしくみ (3: Access to another Application)

- Ticket Granting Ticket を持っている状態で「他のアプリケーション」にアクセスする
  - 有効な Service Ticket が存在しない
  - Service Ticket が Timeout している
  - 異なる “Service Class” へのアクセス
- Ticket Granting Cookie を検証
- 新規 Service Ticket の発行

CAS 認証のしくみ (3: Access to another Application (0))

LDAP Server

Web Application

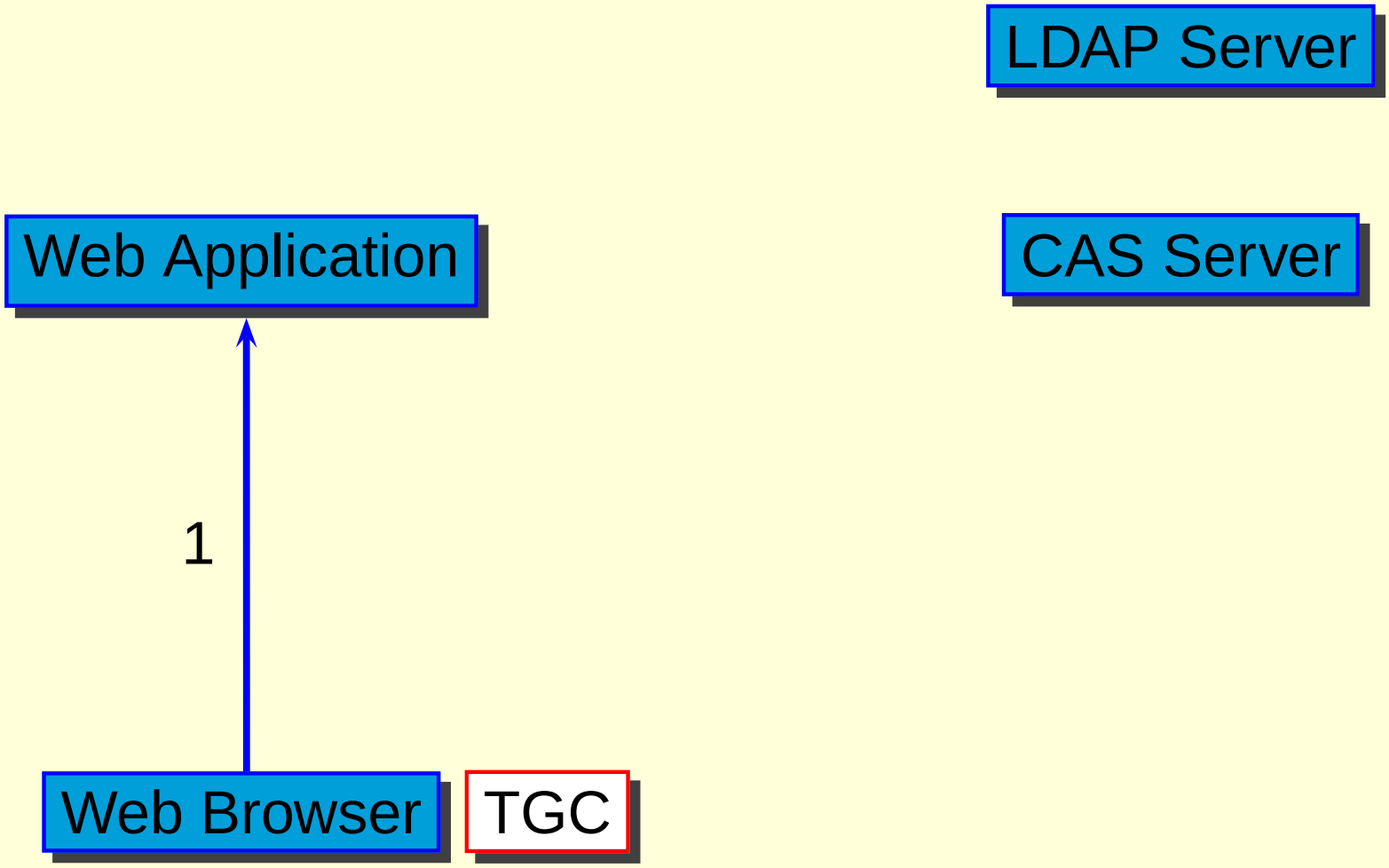
CAS Server

Web Browser

TGC

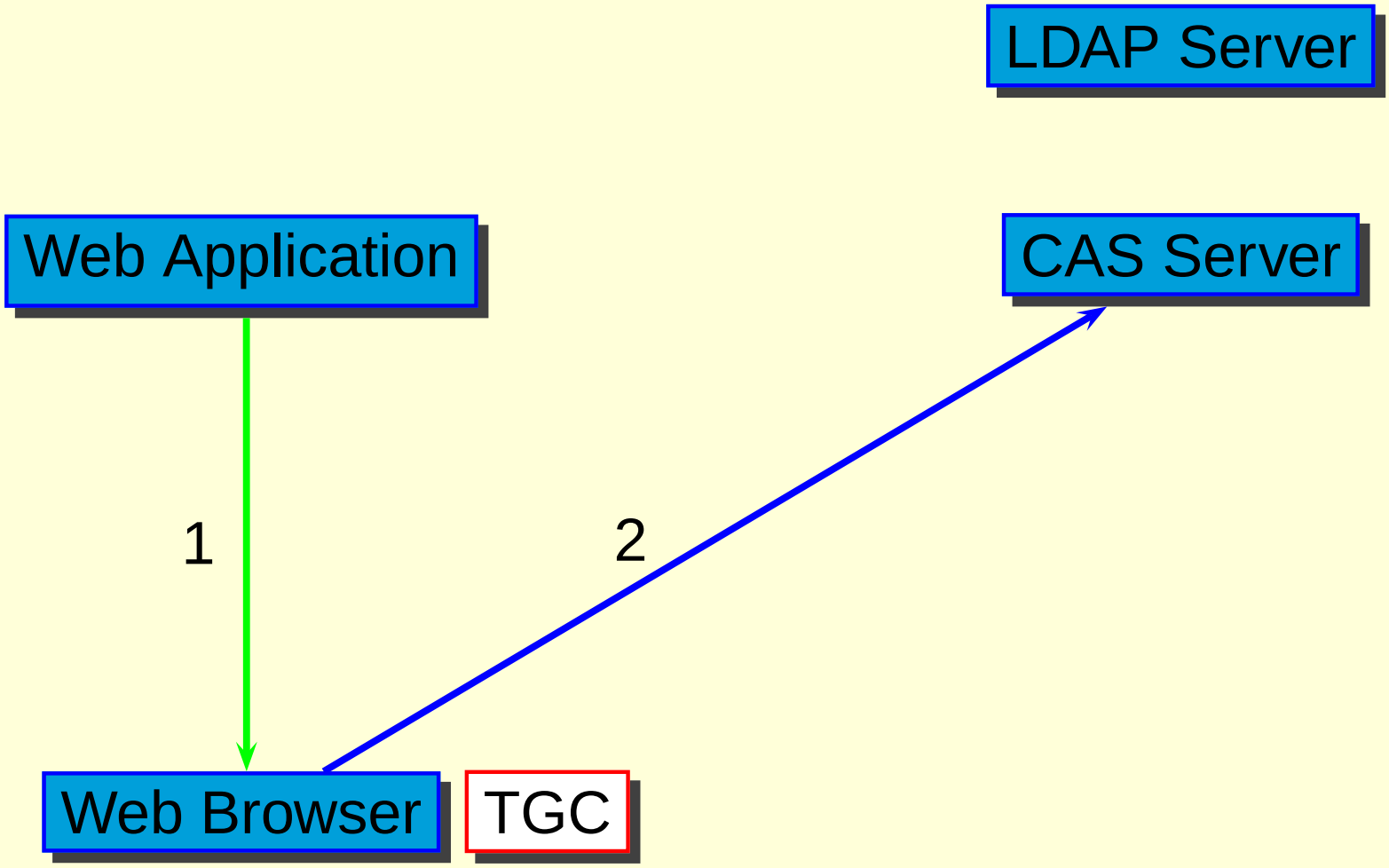
no ST, ST is expired or ST is belonged to different ACCESS CLASS

CAS 認証のしくみ (3: Access to another Application (1))



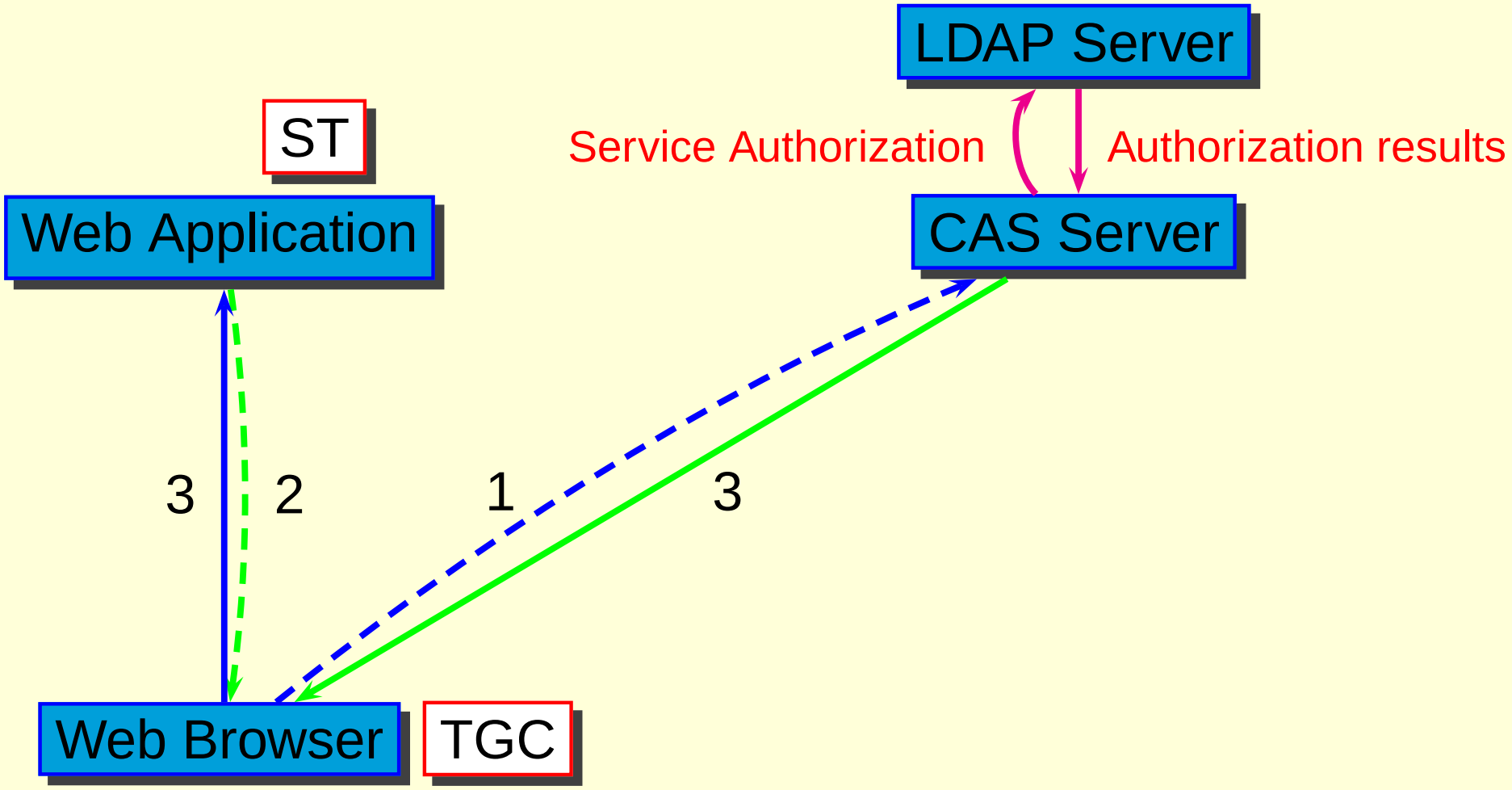
1. Access to **`https://aFQDN/a.html`**

CAS 認証のしくみ (3: Access to another Application (2))



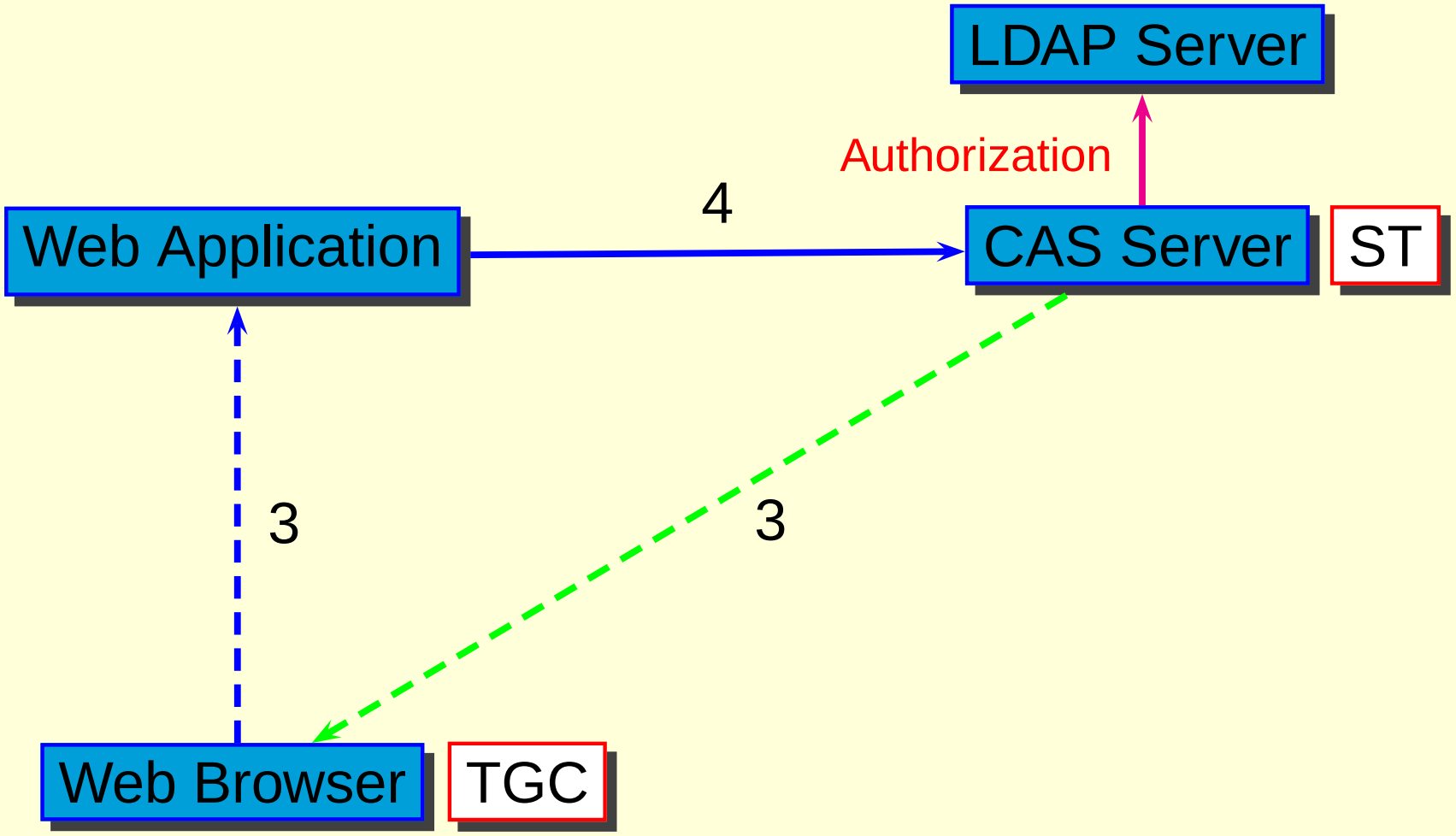
4. Redirect to
   `https://CAS/login&service=https://aFQDN/a.html`

CAS 認証のしくみ (3: Access to another Application (3))



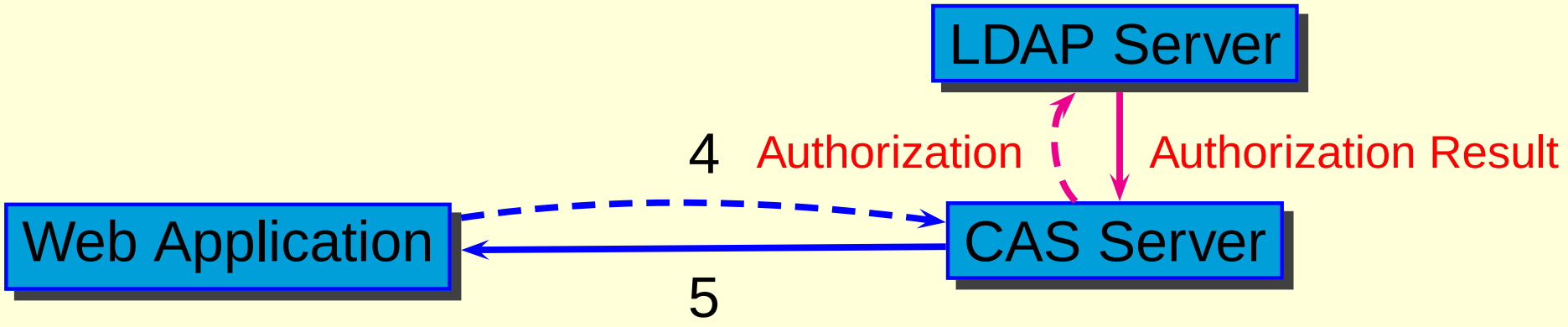
5. Redirect to **`https://aFQDN/a.html&ticket=ST-xxx`**

CAS 認証のしくみ (3: Access to another Application (4))



6. Verify Service Ticket

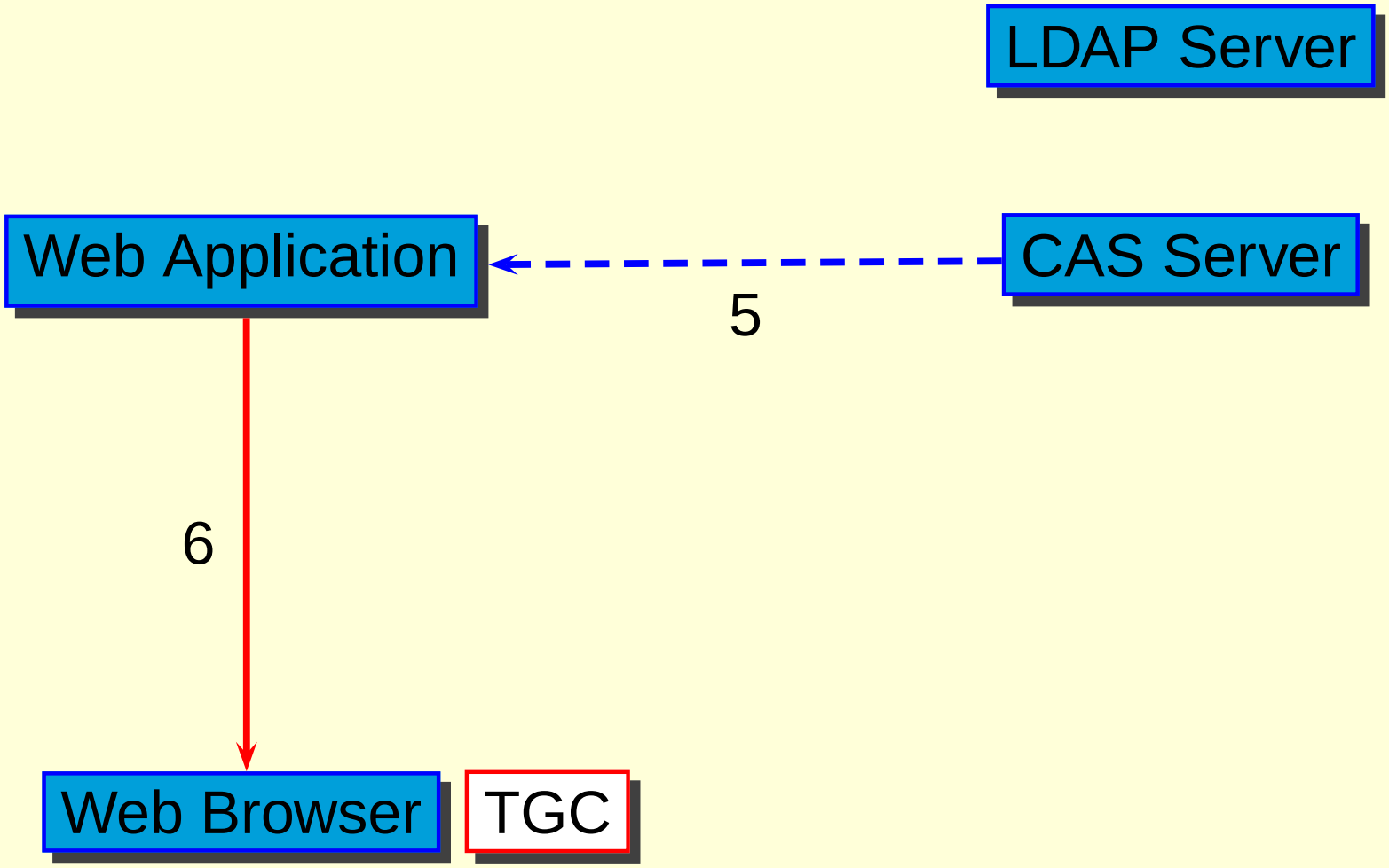
CAS 認証のしくみ (3: Access to another Application (5))



Web Browser TGC

7. Receive verify result form CAS server

CAS 認証のしくみ (3: Access to another Application (6))



8. Receive Data from Application Server

CAS の利点

- Single Sign On 環境を容易に実現できる
- Web Application 側には CAS client module を追加するだけ
- Web Application はユーザのパスワードを受け取らない
- Web Application が認証データベースへ直接アクセスしない
- 軽くて高速に動作
  - 最大アクセス実績： 4000 回/分
    - Sun Fire V480 (1.0GHz UltraSPAC III Cu x 2)
    - 4.0GB Memory
    - Solaris 8

将来にむけて

- クライアント証明書などへの対応
- 非 Web Application での利用方法の開発
- Federated CAS の開発
- CAS Version 3 への対応