2. Service Ticket 服务票据,每个 Service 对应唯一 ST,建立一个 Service Ticket Management,作用:1)通过 ST 拿到 Service,校验该 Service 是否允许;2)通过 ST 拿到 TGT,校验全局会话是否有效。 3.生成 Ticket Granting Ticket 作为全局会话 SESSIONID,映射 User,映射该用户会话下所有服务的 ST。登出时根据 TGT 拿到 ST,通过 ST 拿到 Service,获取到 Logout api。 \equiv 、 Application Service 1. 保存唯一的 ST, 建立局部会话的 SESSIONID 与 ST 的映射, 因为 ST 唯一性, 也可以直接拿 ST 作为会话标识。 三、解释 TGT: Ticket Granted Ticket(俗称大令牌,或者说票根,他可以签发ST,即全局会话的 SESSIONID,存在 CAS 服务器上。) TGC: Ticket Granted Cookie(cookie中的value,存在浏览器的 Cookie 中,根据他可以找到TGT。) ST: Service Ticket(小令牌,由TGT、用户、应用服务生成的,与应用服务——对应,可以作为应用服务器中的 SESSIONID。) 未登录访问 User **Browser Application CAS** -GET www.a.com--www.a.com-1. 从请求携带的cookie中获取 SESSIONID 2. 没有SESSIONID,可能没有 登,即局部会话不存在 3. 校验有没有从CAS获取过 rigisterService 用于注册访问服务(服务地址、登出地址)和 ST, 即是否有认证服务的票据 4. 如果没有ST,则直接重定向 302 Location: www.cas.com/login? 到中心认证服务的登录请求 service=www.a.com -GET www.cas.com?service=www.a.com-1. 在/login路由中获取service值 2. 从全局注册的服务rigisterService中匹配service 3. 如果有注册,则返回登录页面 4.如果没有,则返回报错页面,显示未注册服务 无权限操作 └ - Login Form-–display Login Form--POST www.cas.com/login?service=www.a.com----Submit Login Form---1. 校验 Service 2. 建立全局会话,根据用户 User 生成 TGT,作 为当前用户的全局sessionID 3. 生成单个应用服务票据 Service Ticket 4. 将ST与当前User和当前应用Serveice建立映 Set-Cookie: CASTGC=TGT=12345 射,以ST为key 302 Location www.a.com?ticket=ST-12345 5.对CAS server 的响应以CASTGC设置 set-cookie 6.302重定向到service的url中拼接上该服务对应 -GET www.a.com?ticket=ST-12345-路由在获取ticket值,向 CAS发起ticket有效性校验 GET www.cas.com/serviceVaildate? service=www.a.com?ticket=ST-12345 1.获取ST,从ST生成时ST与Service的映射 中获取service与路由中service是否一致 2. 验证 ST 有效性 --200 -1. 生成局部会话,并设置SESSIONID 2. 建立SESSIONID与ST映射 3. 返回300,并设置cookie Set-Cookie: SESSIONID 302 Location: www.a.com Cookie:SESSIONID GET www.a.com 校验 SESSIONID 登录未过期 访问 Cookie: SESSIONID request source www.a.com/resource Cookie 未过期 校验 SESSIONID 登录过期访问 request source -www.a.com/resource Cookie 已过期 1. 校验 SESSIONID 过期 2. 通过 SESSIONID找到ST 3. 带上ST请求CAS全局会 话是否存在 GET www.cas.com/serviceValidate? service=www.a.com?ticket=ST-12345 1. 获取ST 2. 校验ST与service 3.获取并校验TGT 1. 当前ST仍有效,重新建议局部会话 2. 生成SESSIONID,并绑定ST关系 Set-Cookie: SESSIONID 200 登录另一个 系统 **Application2 CAS** User **Borwser** -www.b.com -GET www.b.com-1. 从请求携带的cookie中获取SESSIONID 2. 没有SESSIONID,可能没有登录,即局 部会话不存在 3. 校验有没有从CAS获取过ST,即是否有 认证服务的票据 4. 如果没有ST,则直接重定向到中心认证 302 Location www.cas.com/login ?Service=www.b.com 服务的登录请求 Cookie CASTGC=TGT-12345 GET www.cas.com/login?service=www.b.com 1. 在/login路由中获取service值 2. 从全局注册的服务rigisterService中 匹配service, 如果没有匹配返回报错 3. 从当前cookie中获取TGT,全局会话 仍生效 -----302 Location www.b.com?ticket=ST-12345 4. 在当前用户下再生成一个ST, 并对 应service b GET www.b.com?ticket=ST-12345-5. 重定向并携带ST 路由在获取ticket值,向 CAS发起ticket有效性校验 GET www.cas.com/serviceValidate? service=www.b.com?ticket=ST-12345 获取ST,从ST生成时ST与Service的映射中 获取service与路由中service是否一致 -----200-1. 生成局部会话,并设置SESSIONID 2. 建立SESSIONID与ST映射 3. 返回302,并设置cookie Set-Cookie: SESSIONID 302 Location www.b.com Cookie SESSIONID www.b.com 校验 SESSIONID --- display page -主动登出 **Application CAS** User Borwser Cookie: SESSIONID -Logout-GET www.a.com/logout 302 Location www.cas.com/logout? service=www.a.com?ticket=ST-12345 Cookie: CASTGC=TGT-12345 1. 获取 TGT GET www.cas.com/logout?service=www.a.com?ticket=ST-12345 2. 从TGT获取用户所有ST 3. 从ST遍历出所有service 4. 根据rigisterService中查找对应service的登出 地址,并拼接上ST 5. 销毁所有ST,以及全局会话TGT失效 -GET www.a.com/logout?Ticket=ST-12345-6. 返回登录接口,并拼上登出的service 1. 通过ticket拿到SESSIONID 2. 让SESSIONID失效,删除 与ST映射关系 3. 其它应用一样操作 - -200 --302 Location: www.cas.com/login?service=www.a.com Cookie: CASTGC=TGT-123345 GET www.cas.com/login?Service=www.a.com 全局会话失效, 返回登录页 -CAS Login form -←----Login Form----全局会话过期 1. 监听到全局会话 TGT 过期 2. 从TGT获取用户和所有ST 3. 从ST获取对应的service 4. 根据rigisterService中查找对应service的 登出地址,并拼接上ST 5. 销毁所有ST,以及使全局会话TGT失效 -GET www.a.com/logout?ticket=ST-12345-所以 CAS 服务中要有 TGT 与 User 及 1. 通过ticket拿到SESSIONID Sserver的表 2. 让SESSIONID失效,删除 并且每个 Server 有对应的 ST 和 logout 的 与ST映射关系 3. 其它应用一样操作

1. rigisterServer 注册授权服务,也可以建立一个 Service Management,包括 Server Host、User、Server Logout API。作用:1)校验Service是否允许,2)登出时获取 Service Logout api 通知服务销毁局部会话。

映射关系: 一、CAS Server