# 1、单点登录第一种情况(网站顶级域名相同)

如果说顶级域名相同的话，SSO就可以使用cookie加上redis共同的实现，因为在后端是使用的缓存中间件redis，整个在集群或者是分布式系统的环境之下，用户的会话在后端是可以获得的。另外对于cookie，由于顶级域名是相同的，所以他的cookie在所有的产品线，在所有的域名下共享的，这种时候是完全可以使用这种方式，实现单点登录。

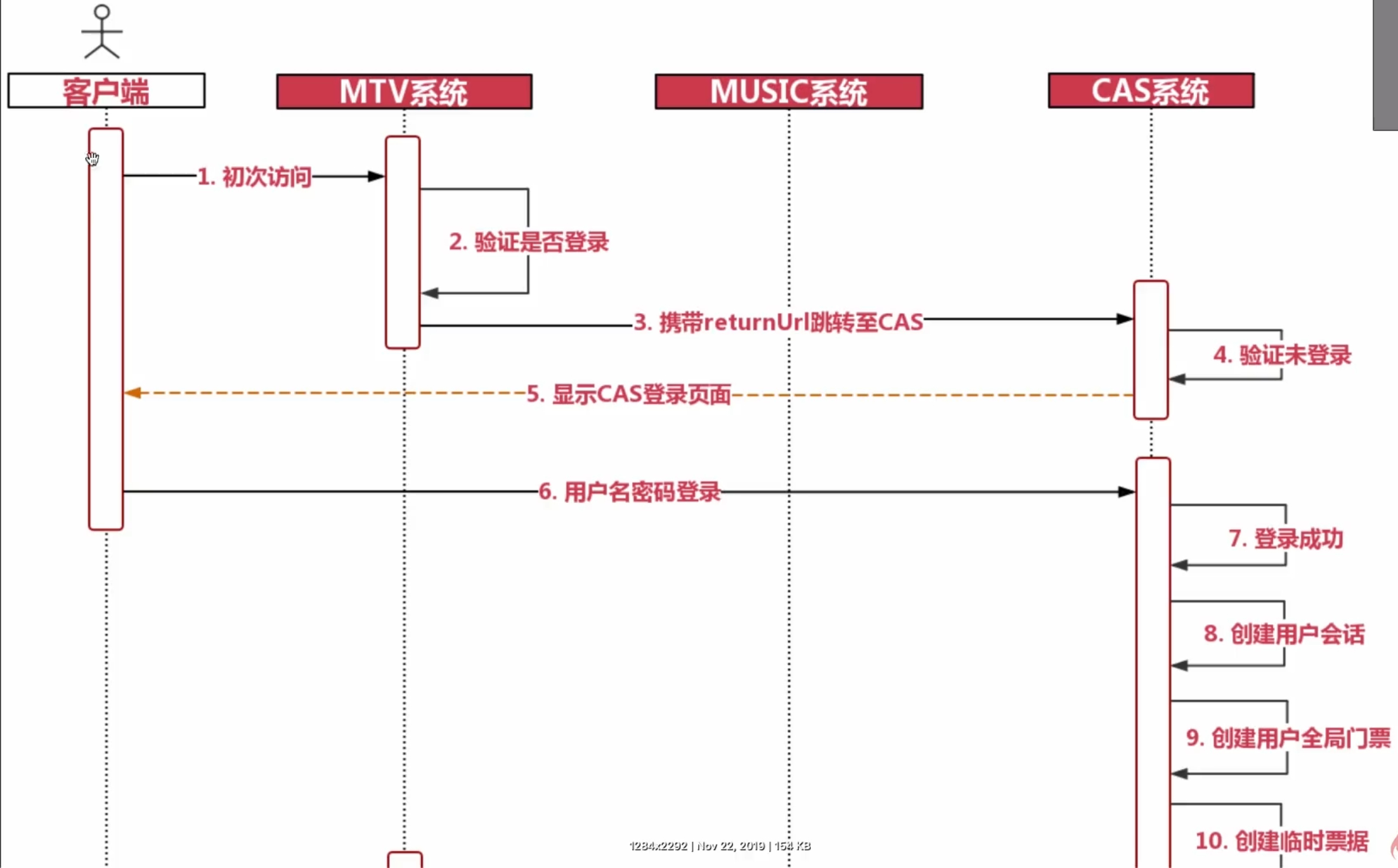
原理也就是分布式会话。

# 2、单点登录第二种情况(顶级域名不同)

如果说现在是有两个不同的网站，甚至是多个的话，他们也是能够去做到单点登录的。其实也是可以的。假设现在是有两个网址www.mtv.com，www.music.com，这是两个不同的网站，是可以通过另外的一种手段去实现单点登录。

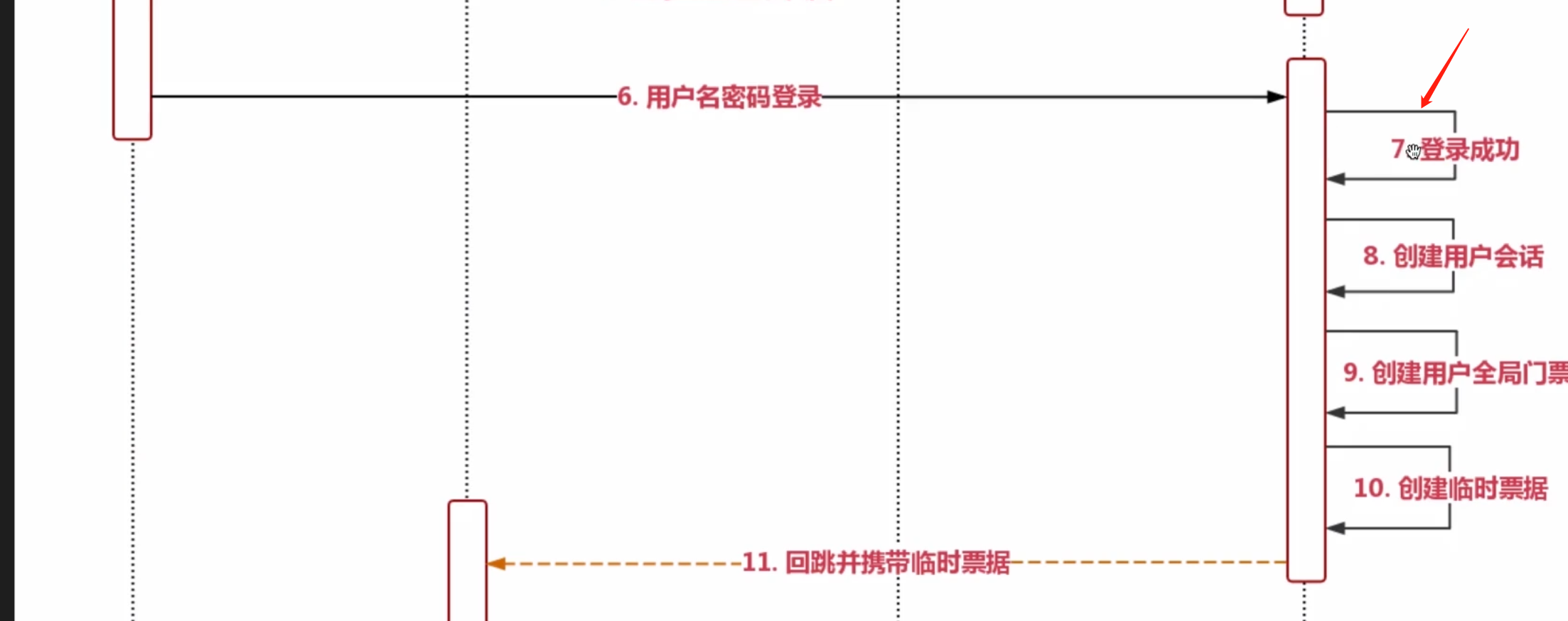
这个时候就需要借助CAS这样一个系统了，全称就是Center Authentication Service，他其实是一种中央认证服务，可以把会话信息或者是说整个全局的token存储在CAS里面，随后当其他系统发起调用的时候，可以通过CAS去做一个中转，他其实就是相当于是一个中介或者说是一个中转站。

# 3、CAS实现跨域名单点登录时序图(用户从来没有登录)



首先在用户访问浏览器的时候，如果说用户在所有的系统里面从来都是没有登录过得话，假设现在第一次是访问的mtv网站，然后MTV网站会去验证一下，验证用户是否是登录了，如果是没有登录的话，MTV系统会携带相应的参数请求到CAS，这个参数其实就是returnUrl，携带的这个returnUrl参数主要是用于去做请求的会跳，因为现在是存在多个系统，所以这个时候，如果说CAS收到你的验证不通过请求并且是处理完毕之后，是要去做一个回调的，回调是基于returnUrl的，然后再CAS这一端的话，肯定也是要验证一下，这个用户他有没有登录，很明显它是没有登录的。这个时候CAS系统就会把登录页面展示给用户，让用户在浏览器去输入用户名和密码去做一个登录。

登录的话，其实一些相应的登录的判断，其实是要在CAS的后端去做一个判断的。



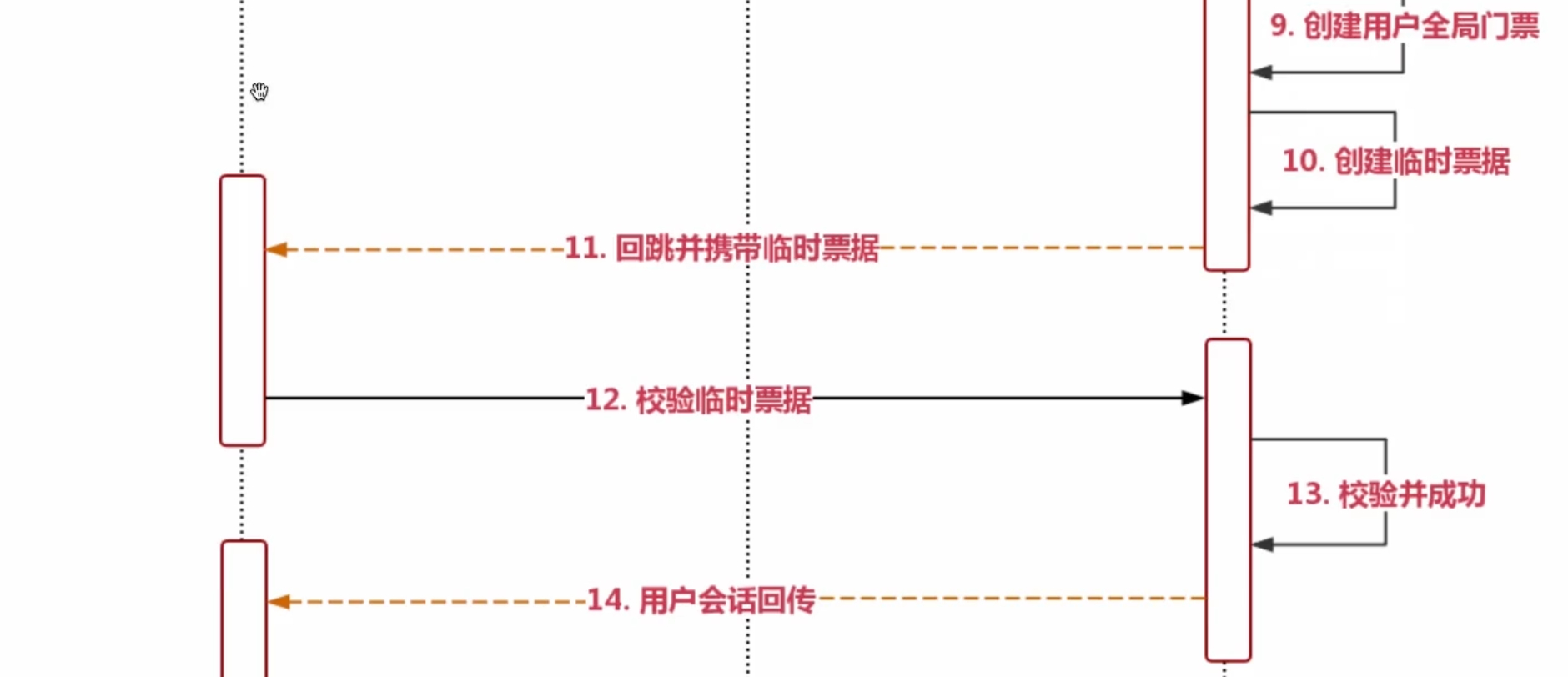
也就是在第七部，假设是登陆成功了，登陆成功之后有几步操作是需要去做的，第一个是创建用户的会话，这个其实就是用户的分布式会话，这个其实就是用户的分布式会话，之前是有操作过的，把用户的会话给创建出来，

最后第八步，创建用户的全局门票，这个其实也就是生成了一个token。

生成了一个token之后，作为一个用户的全局门票，保存在CAS这一端的，他会保存在cookie里面，他也会在Redis和用户会话也就是用户的ID做一个关联的绑定，就是代表我当前这个用户在我们的全局的范围内，在CAS系统里面，是做了一个登录的。所以这个是第九步所做的一个操作。

随后第十步，这一步是要为用户创建一个临时的票据。这个临时的票据创建完了之后，是给MTV站点去使用的。这个时候用户要去做登录了嘛。

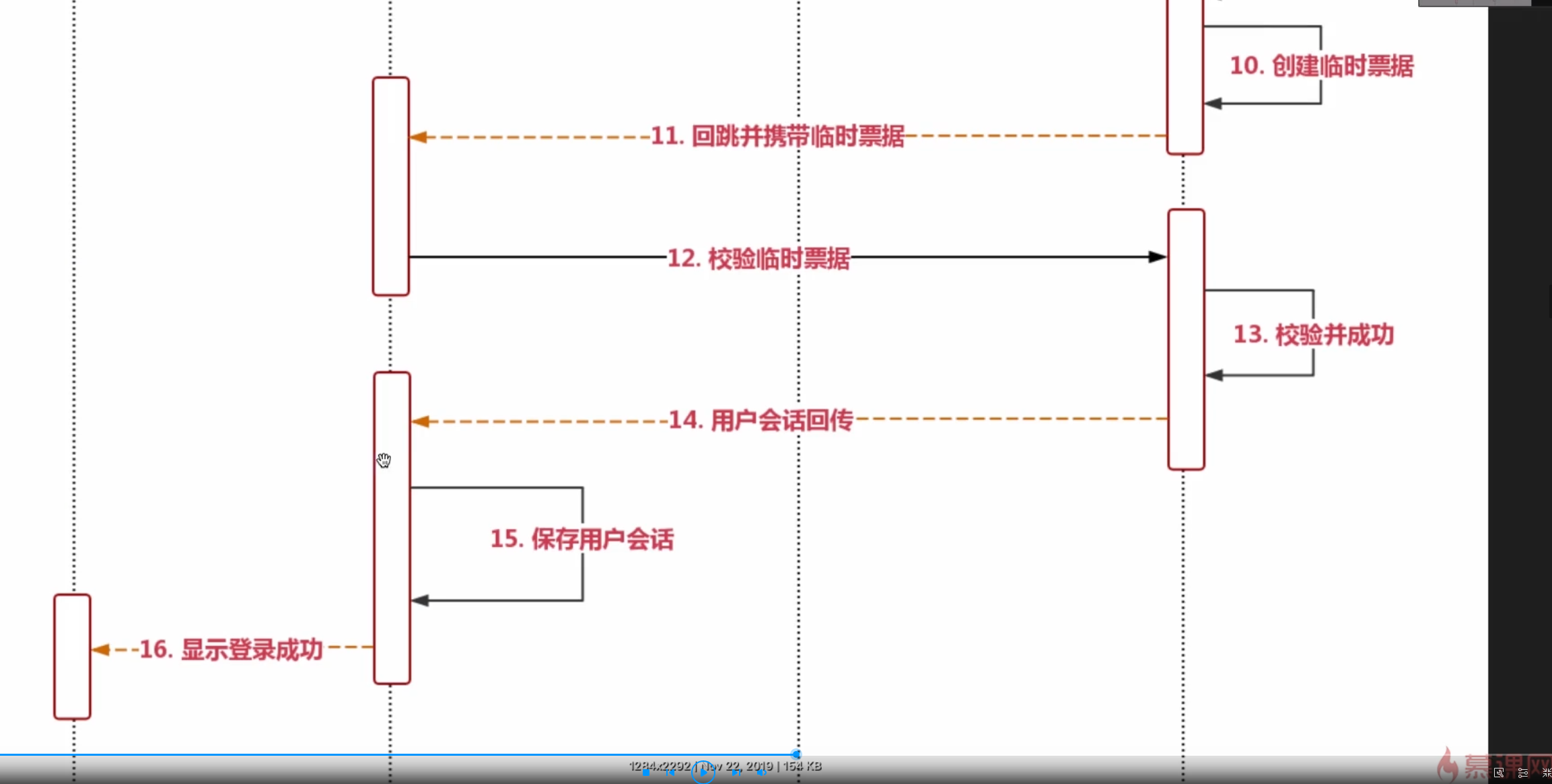
所以在第十一步，CAS创建的临时票据就会回调到MTV系统。



回调的同时CAS系统会携带上临时票据，这个临时票据其实就是CAS系统颁发给他的。

之后在第十二步是要去做一个校验，校验的时候，它是鞋带上了这个临时的票据，再一次的发起请求到CAS，让CAS系统去做一个验证，如果验证成功，判断一下他这个临时的票据在CAS系统里面是存在的，也就是校验成功了，所以这个时候MTV系统就可以获取在CAS系统里面的用户会话。

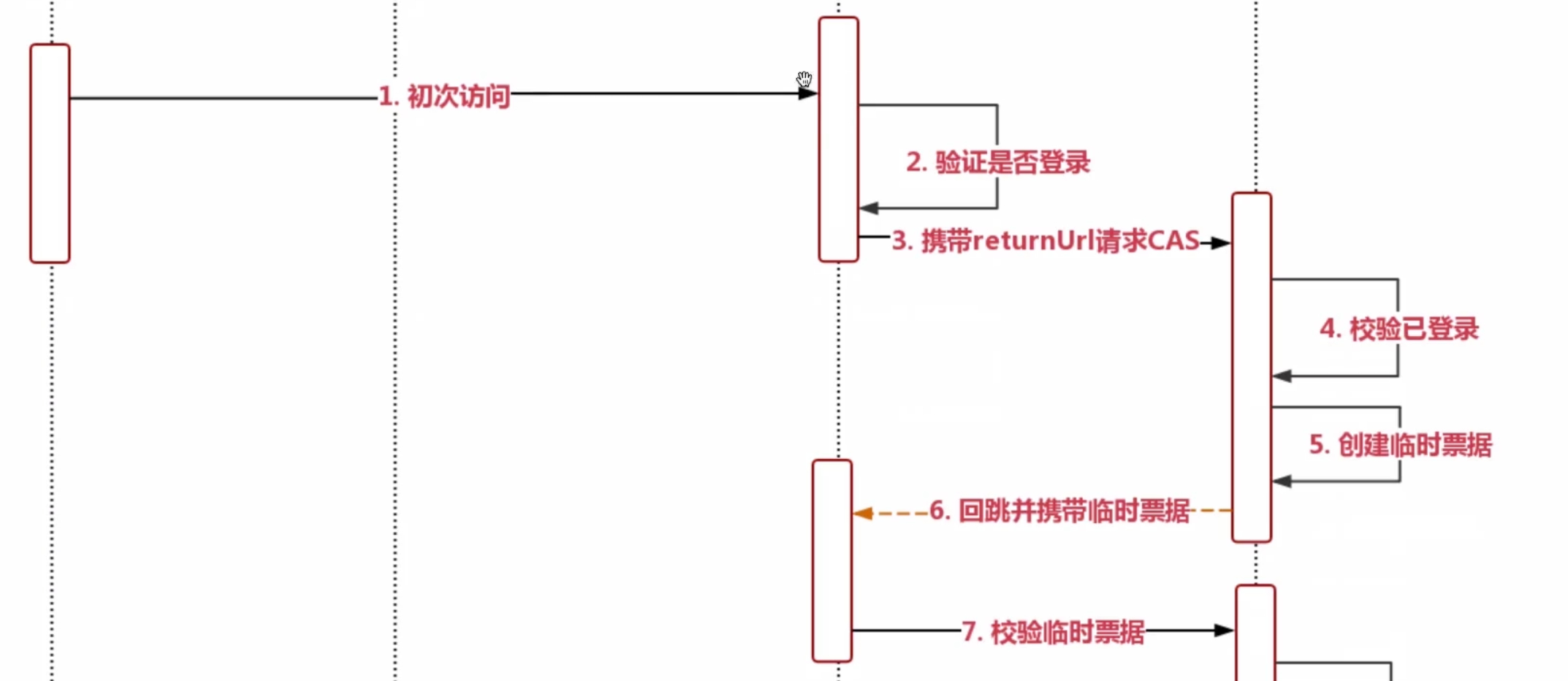
查询到用户会话了以后。



第十四步，会将用户的会话回传到MTV系统，那么在MTV的站点上就可以拿到会话了。

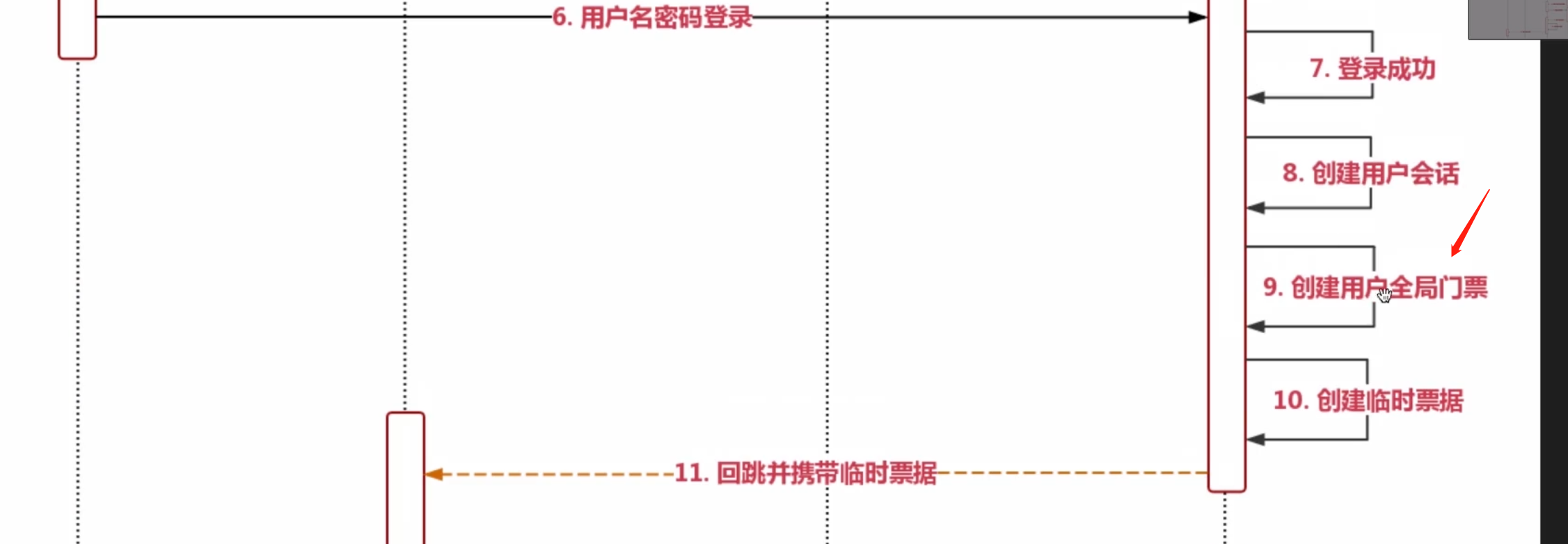
在第十五步骤的时间，保存用户会话的话是保存在前端，也就是在浏览器的cookie里面，这样子在站点上，在网页里面，就可以展示说欢迎xxx登陆。目前已经是成功的登陆了。

# 4、CAS实现跨域名单点登录时序图(用户登陆过了)

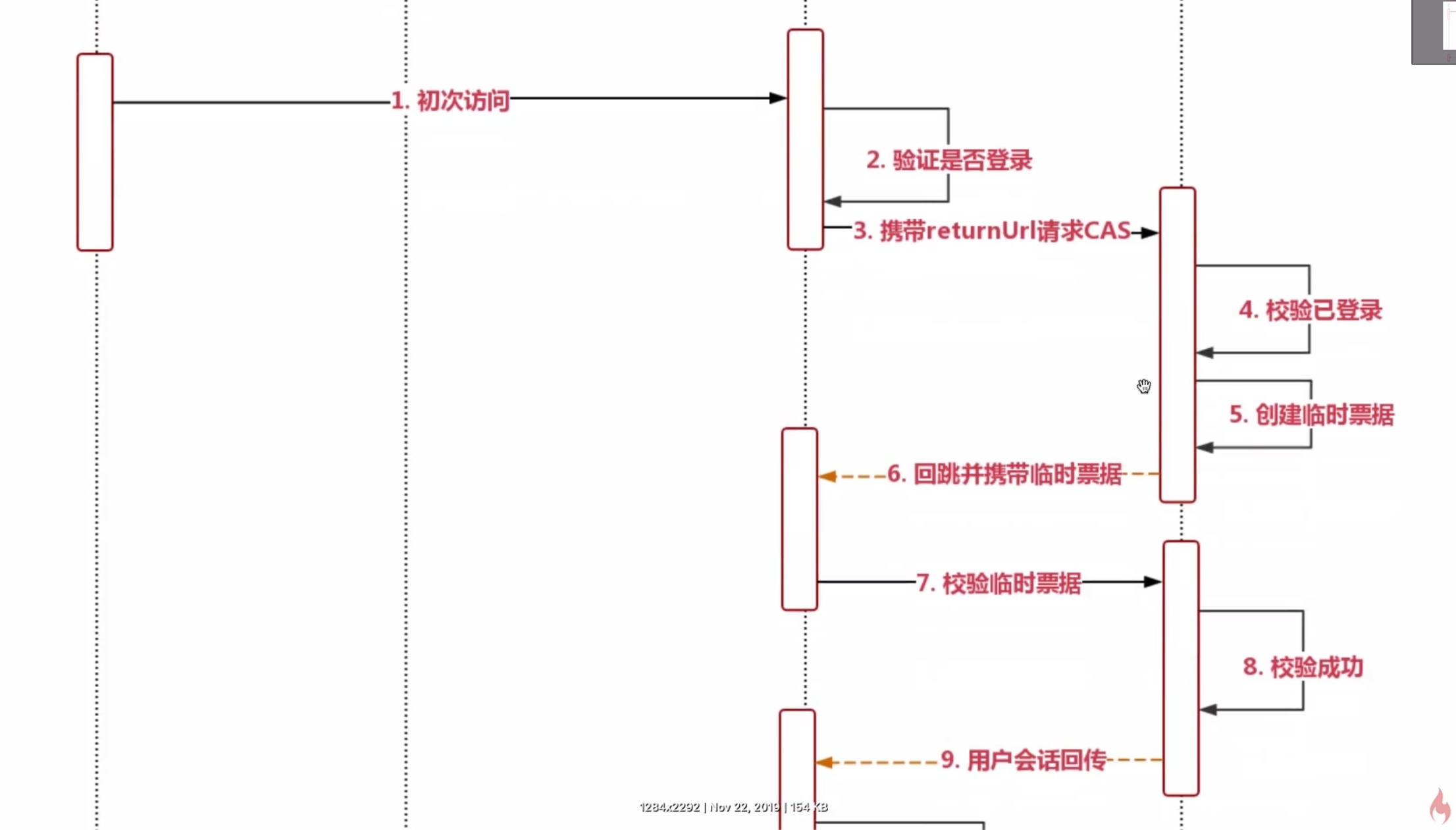


首先用户也是初次访问了另外一个系统，music系统(刚在mtv系统登陆过了)

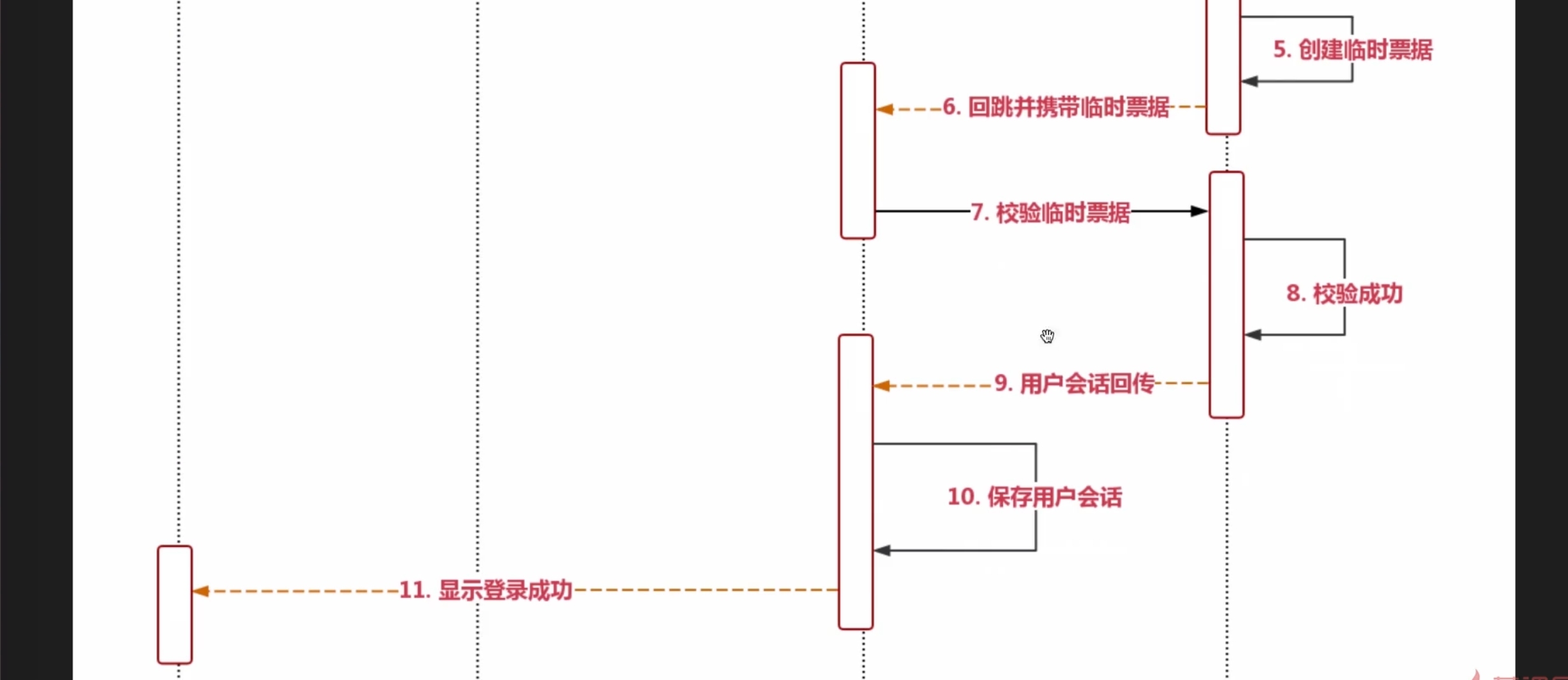
首先第一步也是初次访问到了music站点，验证一线是不是登录过了，很明显是没有登录的，随后携带music站点的returnurl请求CAS系统，这个时候他会发现其实在CAS系统已经是登录过了，进行一个校验的，因为在MTV系统登录的时间已经是创建一个用户的全局门票了。



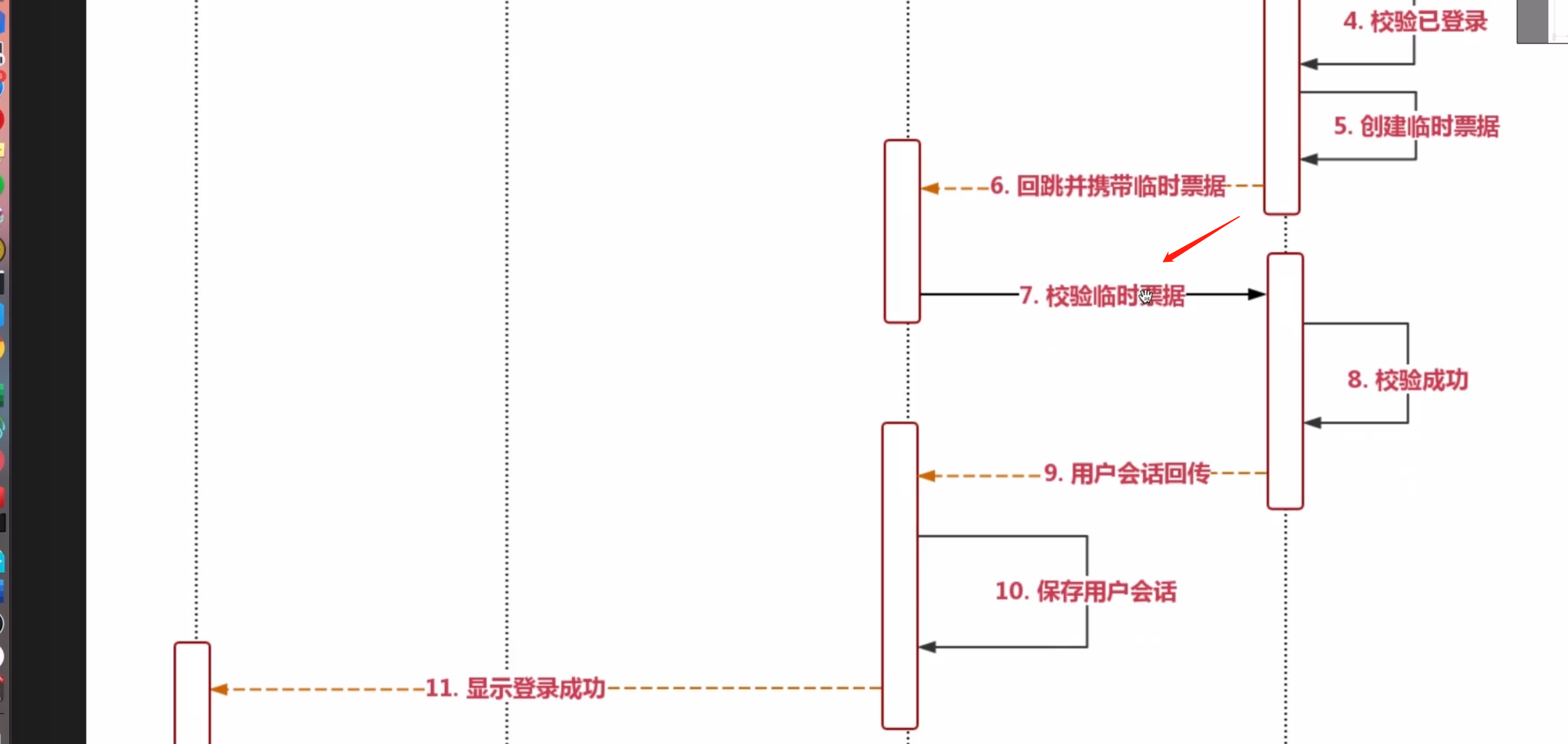
只要是这个全局门票是存在的，所有用户的会话信息就是可以通过CAS系统获取到的



拿到全局门票之后，就代表你的这个请求是不需要再去做二次登陆了，通过CAS系统就直接是可以创建一个临时的票据，创建了临时票据之后，再一次的回跳携带返回给music站点，下面步骤其实就是music系统携带者临时票据在一次请求CAS系统校验用户数据，校验成功之后，把用户的会话回传给music系统。



然后在music系统的前端站点去做保存，然后展示登陆成功，就是这样一个流程。



这边做CAS系统校验的时候因为是临时的票据，临时的票据在使用之后是要被删除的，因为临时票据是一次性使用的。

在保存用户会话步骤的时间，它仅仅只是在cookie里面去吧用户会话的信息去做一个保存，因为用户的会话，由于在后端使用的是一个redis，只要是把用户的userId以及是会话里面的token携带放到header里面传过去，在后端是可以获得用户的整个信息的。所以保存用户会话在前端保存就可以了。