Technical Disclosure Commons

Defensive Publications Series

April 2020

SUPERVISED GUEST BROWSER ACCESS TO SPACES AND MEETINGS

Owen Friel

Ollie Fagan

John Costello

SUPERVISED GUEST BROWSER ACCESS TO SPACES AND MEETINGS

AUTHORS:
Owen Friel
Ollie Fagan
John Costello

ABSTRACT

Techniques are described for a browser authorization flow to enable guest onboarding and authorized access to a service. These techniques allow authorized users to create persistent guest accounts by simply scanning a code rendered on a guest browser. There is no need for the guest user to install an application on their smartphone (or even to have a smartphone), to have access to their email, or to click on any email verification links.

DETAILED DESCRIPTION

Inviting guest users to spaces and meetings can be a difficult and time-consuming process that relies on emailing invitations, clicking email links, online registration, and password setting. Research has shown that the more frictionless the user experience, the more likely the user is to accept it. Therefore, it is desirable to have a solution for guest users who want to use a laptop browser instead of a smartphone application to access a service.

Figures 1-4 below collectively illustrate an example call flow. In this example, guest user Bob is visiting Alice at her company. Alice wants to invite Bob to collaborate using a service. Bob wants to join the service via the browser on his laptop rather than an application on his smartphone. For example, Bob may want to share documents or content stored on his laptop. Furthermore, Bob has internet access from his laptop but does not have access to his email account (e.g., Bob does not have Virtual Private Network (VPN) access to his corporate email service), and as such is not able to click on an email validation link.

As shown, Bob points his browser to the service, which may in turn prompt Bob for an email. If required, Bob enters his email. A guest activation code is then rendered on the browser (e.g., as a QR code). If an email was requested and provided, the email is embedded within the code. The code includes a secure random element.

The service creates a pending guest account that is bound to the code and to the browser session. Alice is an authorized user and is logged into the service. Alice scans the code on Bob's browser (e.g., the QR code), or manually enters the code (e.g., if it is rendered as an alphanumeric string). If the code included an embedded email, then Alice may have to accept the email before proceeding. If the code did not include an embedded email, then Alice has to enter the guest email.

Alice's client presents the code and email (whether embedded in the code or explicitly provided by Alice) to the service, instructing the service to authorize access for the browser session. The service confirms the code is valid and proceeds to create the guest account, bind it to the email, and issue tokens to the browser session. An email verification is then sent to Bob's email with a callback Uniform Resource Identifier (URI) embedded in the email. There is no need for Bob to verify the email yet.

Bob's browser persists the tokens and can now access the service as a verified guest user. Any content or messaging that happens between Bob and the service is persisted against Bob's account. Bob can continue to use the browser session for as long as the tokens last, which may be some service-specified timeout (e.g., weeks or months). The service may enforce policy restrictions on the operations that Bob can perform until such time as Bob verifies his email. Once Bob verifies his email address, which could be days, weeks, or even months later, the service based on policy may grant Bob access to additional features.

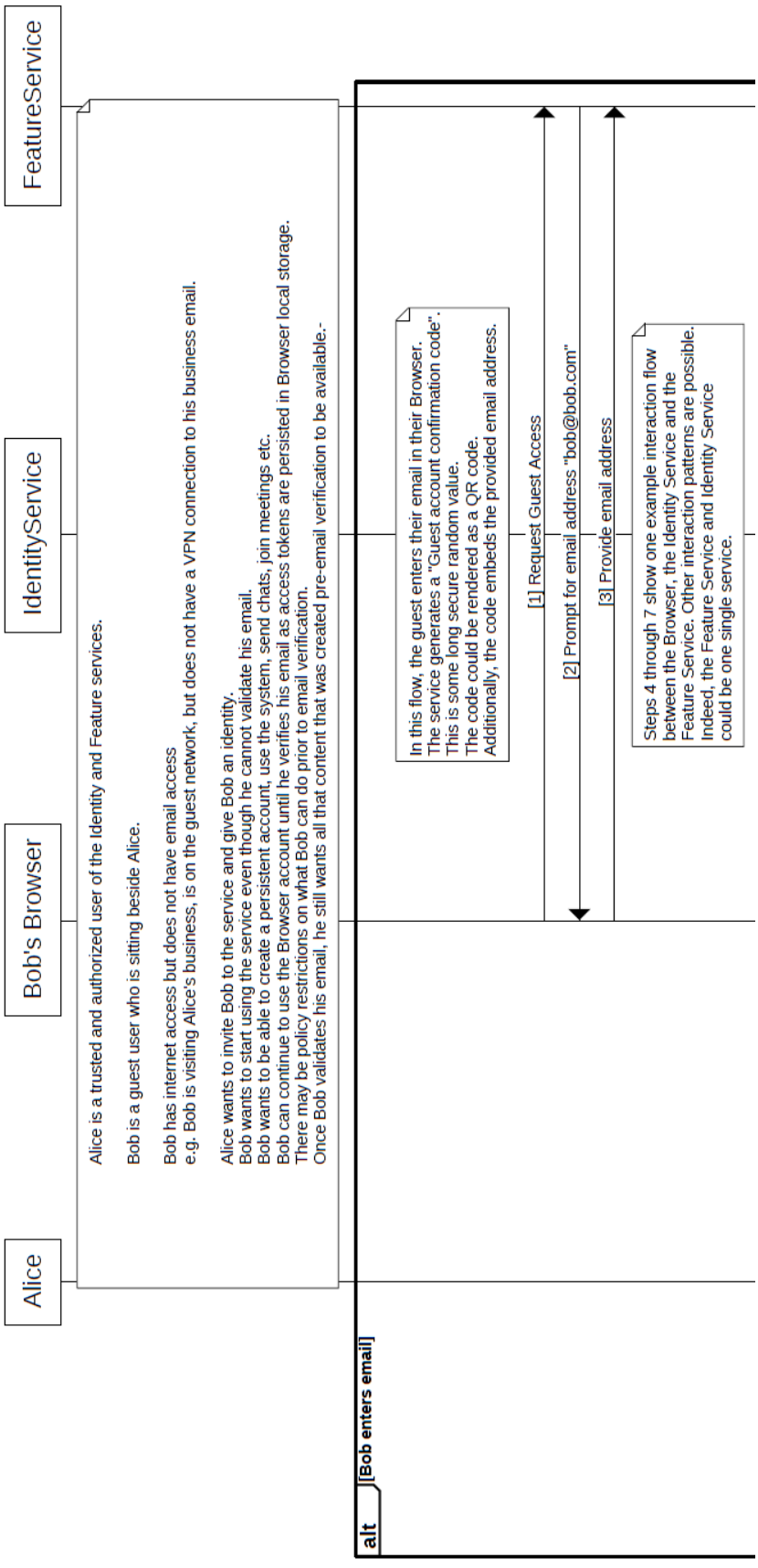
Alice
Bob's Browser
IdentityService
FeatureService
Alice is a trusted and authorized user of the Identity and Feature services.
Bob is a guest user who is sitting beside Alice.
Bob has internet access but does not have email access
e.g. Bob is visiting Alice's business, is on the guest network, but does not have a VPN connection to his business email.
Alice wants to invite Bob to the service and give Bob an identity.
Bob wants to start using the service even though he cannot validate his email.
Bob wants to be able to create a persistent account, use the system, send chats, join meetings etc.
Bob can continue to use the Browser account until he verifies his email as access tokens are persisted in Browser local storage.
There may be policy restrictions on what Bob can do prior to email verification.
Once Bob validates his email, he still wants all that content that was created pre-email verification to be available.-
alt
[Bob enters email]
In this flow, the guest enters their email in their Browser.
The service generates a "Guest account confirmation code".
This is some long secure random value.
The code could be rendered as a QR code.
Additionally, the code embeds the provided email address.
[1] Request Guest Access
[2] Prompt for email address "bob@bob.com"
[3] Provide email address
Steps 4 through 7 show one example interaction flow
between the Browser, the Identity Service and the
Feature Service. Other interaction patterns are possible.
Indeed, the Feature Service and Identity Service
could be one single service.

Figure 1

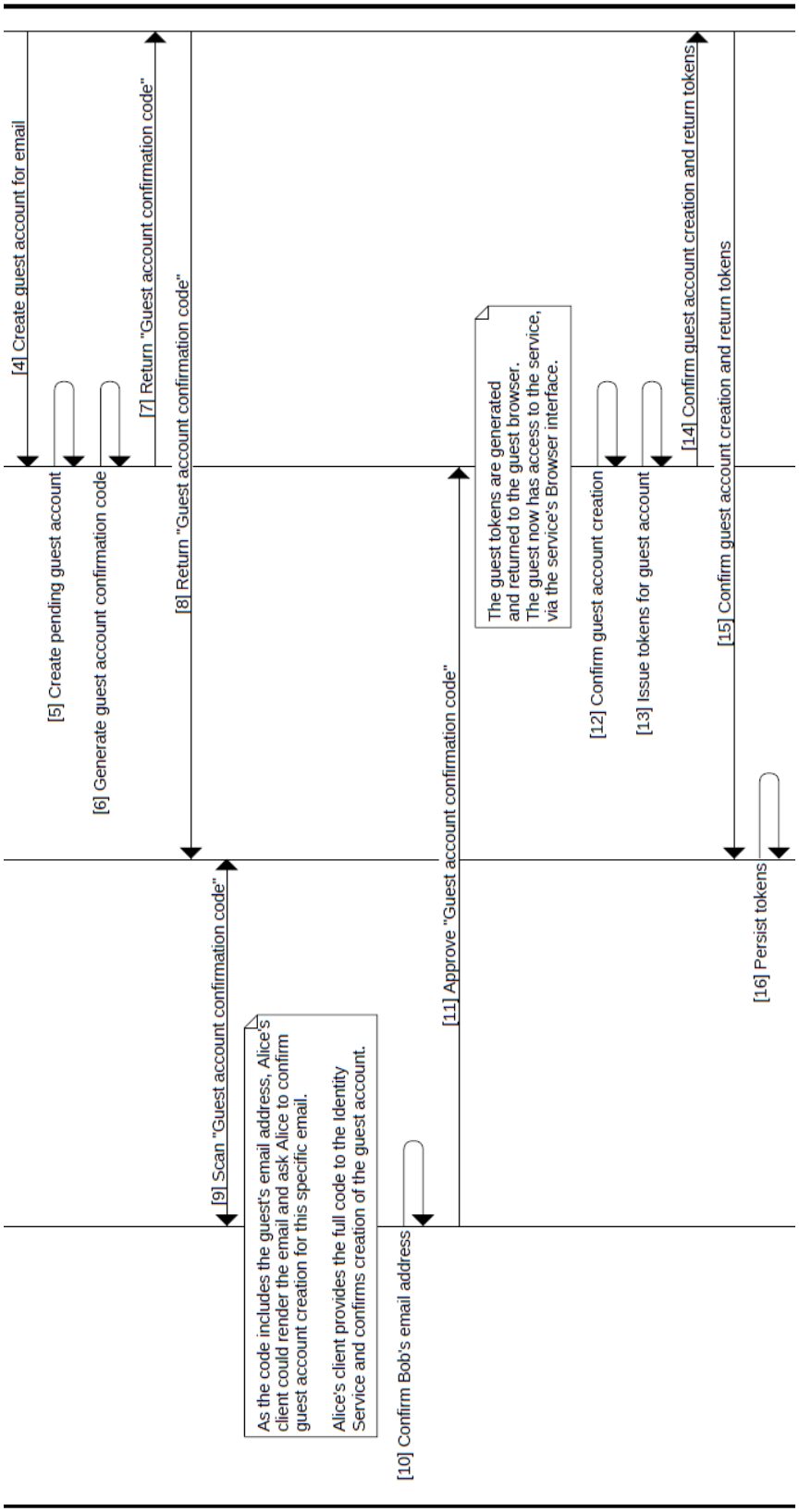
[4] Create guest account for email
[5] Create pending guest account
[6] Generate guest account confirmation code
[7] Return "Guest account confirmation code"
[8] Return "Guest account confirmation code"
[9] Scan "Guest account confirmation code"
As the code includes the guest's email address, Alice's client could render the email and ask Alice to confirm guest account creation for this specific email.
Alice's client provides the full code to the Identity Service and confirms creation of the guest account.
[10] Confirm Bob's email address
[11] Approve "Guest account confirmation code"
The guest tokens are generated and returned to the guest browser. The guest now has access to the service, via the service's Browser interface.
[12] Confirm guest account creation
[13] Issue tokens for guest account
[14] Confirm guest account creation and return tokens
[15] Confirm guest account creation and return tokens
[16] Persist tokens

Figure 2

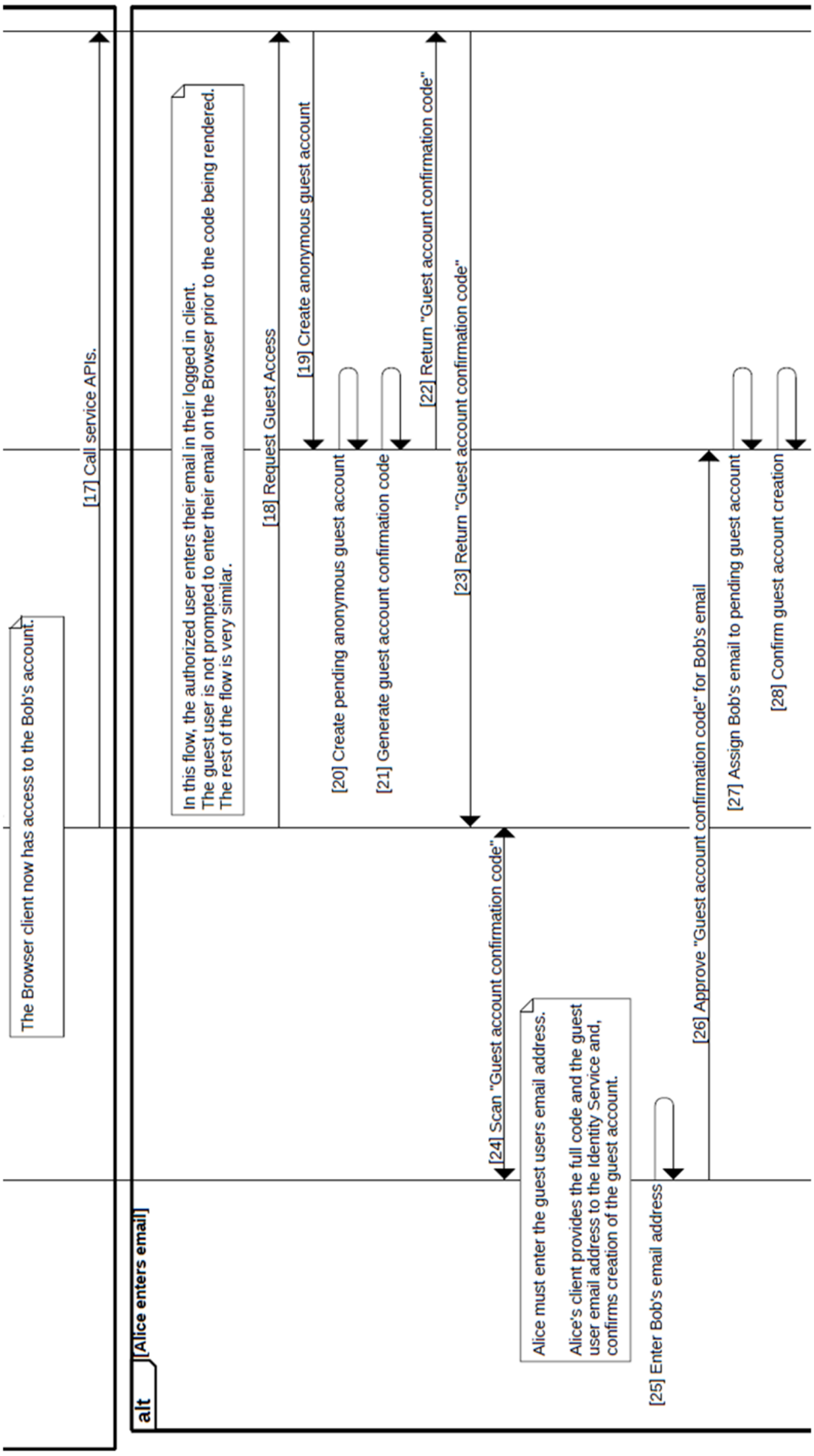
The Browser client now has access to the Bob's account.
[17] Call service APIs.
In this flow, the authorized user enters their email in their logged in client.
The guest user is not prompted to enter their email on the Browser prior to the code being rendered.
The rest of the flow is very similar.
[18] Request Guest Access
[19] Create anonymous guest account
[20] Create pending anonymous guest account
[21] Generate guest account confirmation code
[22] Return "Guest account confirmation code"
[23] Return "Guest account confirmation code"
alt
[Alice enters email]
[24] Scan "Guest account confirmation code"
Alice must enter the guest users email address.
Alice's client provides the full code and the guest user email address to the Identity Service and, confirms creation of the guest account.
[25] Enter Bob's email address
[26] Approve "Guest account confirmation code" for Bob's email
[27] Assign Bob's email to pending guest account
[28] Confirm guest account creation

Figure 3

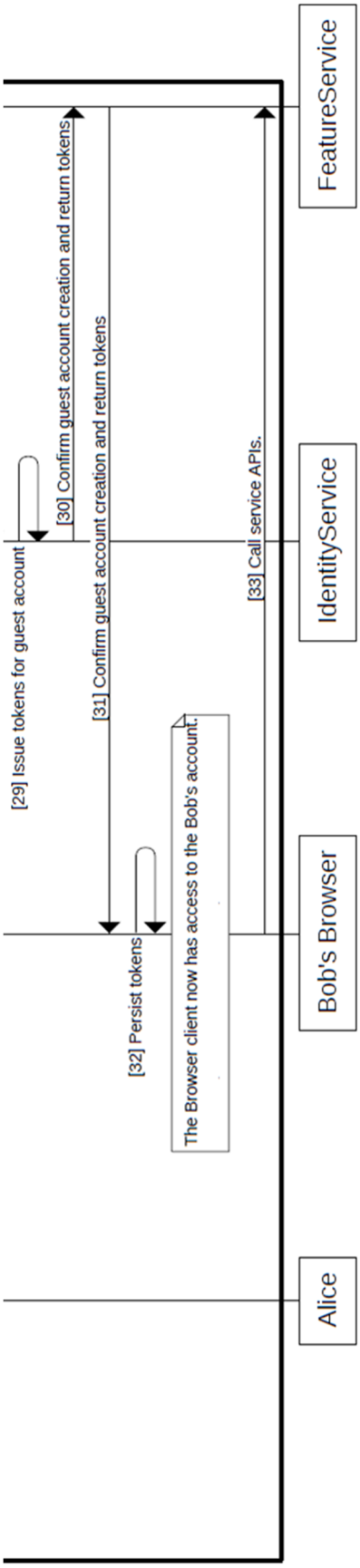
FeatureService
IdentityService
Bob's Browser
Alice
[29] Issue tokens for guest account
[30] Confirm guest account creation and return tokens
[31] Confirm guest account creation and return tokens
[32] Persist tokens
The Browser client now has access to the Bob's account.
[33] Call service APIs.

Figure 4

The techniques described herein provide a long lived persistent authorized identity for a guest user, where the guest user does not have a mobile phone or Near Field Communications (NFC) enabled client machine, and instead is using a basic browser on a laptop or desktop machine. A browser QR may be used to create a persistent long-lived authorized identity for a guest, rather than for oneself. The simple mechanism described herein may authorize a new user and give the new user a persistent long-lived identity, even when the user only has a browser on their laptop.

In summary, techniques are described for a browser authorization flow to enable guest onboarding and authorized access to a service. These techniques allow authorized users to create persistent guest accounts by simply scanning a code rendered on a guest browser. There is no need for the guest user to install an application on their smartphone (or even to have a smartphone), to have access to their email, or to click on any email verification links.