Identity management

The step after (and before) device management

Hello and welcome to this talk about identity management!

My name is Yoann Gini, I run a small consultancy company in France, delivering services for SMB and EDU all around the word.

Yoann Gini

CEO

World Wide Delivery

Externalized CTO/CIO

abelionni

Auditing

Mass Deployment Consultant

Apple Consultant Network

Yoann Gini

System & Network Administrator

As a system and network administrator, I work a lot on topics related to macOS, security and scaling.

You can usually find my in the usual suspects for topics like Security, Network Architecture, SmartCard Services, Reverse Engineering and Hacking.

Yoann Gini

System & Network Administrator

Security

Network Architecture

SMB Information System

SmartCard Services

Reverse Engineering

Hacking

Yoann Gini

Software Developer

I’m also a hobbyist software developer. I’ve created tools like Hello IT, ARD Inspector, Mobile Certificates and Radius/VPN Admin Tools.

Yoann Gini

Software Developer

Mobile Certificates

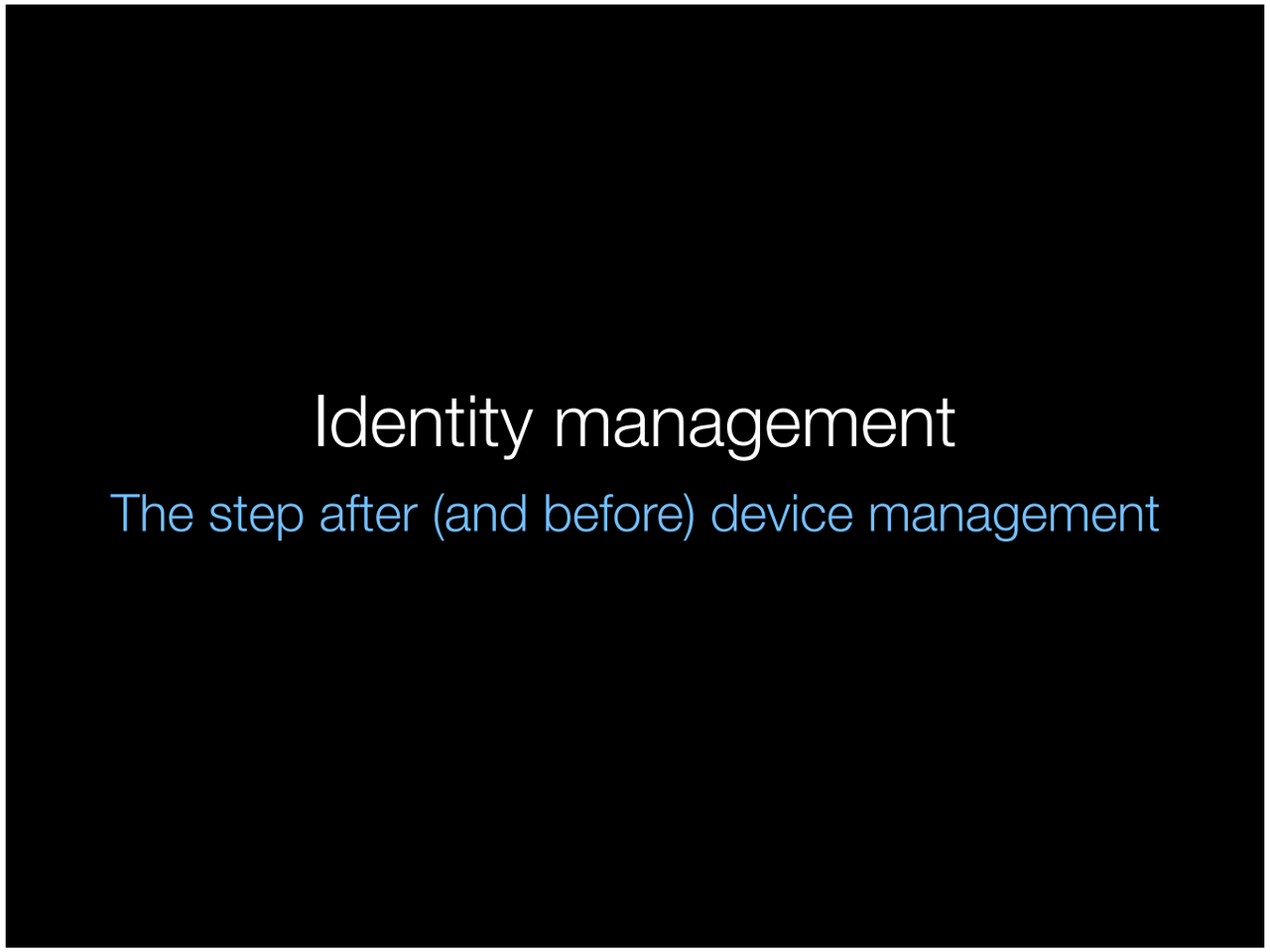
ARD Inspector

Hello IT

Radius Admin Tools

VPN Admin Tools

DockServiceManager



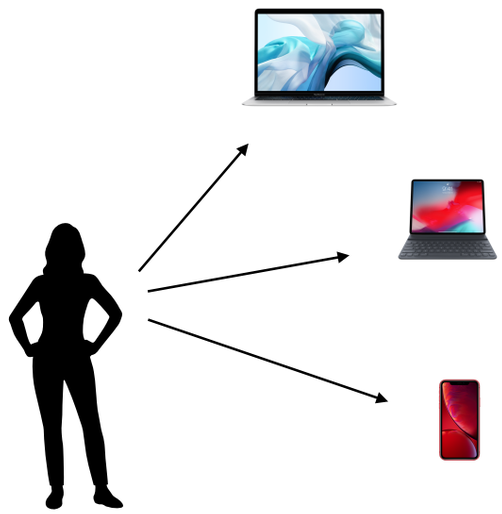
Our time today will be dedicated to identity management and why I consider it as the main step after device management, or even before when you get things done the right way!

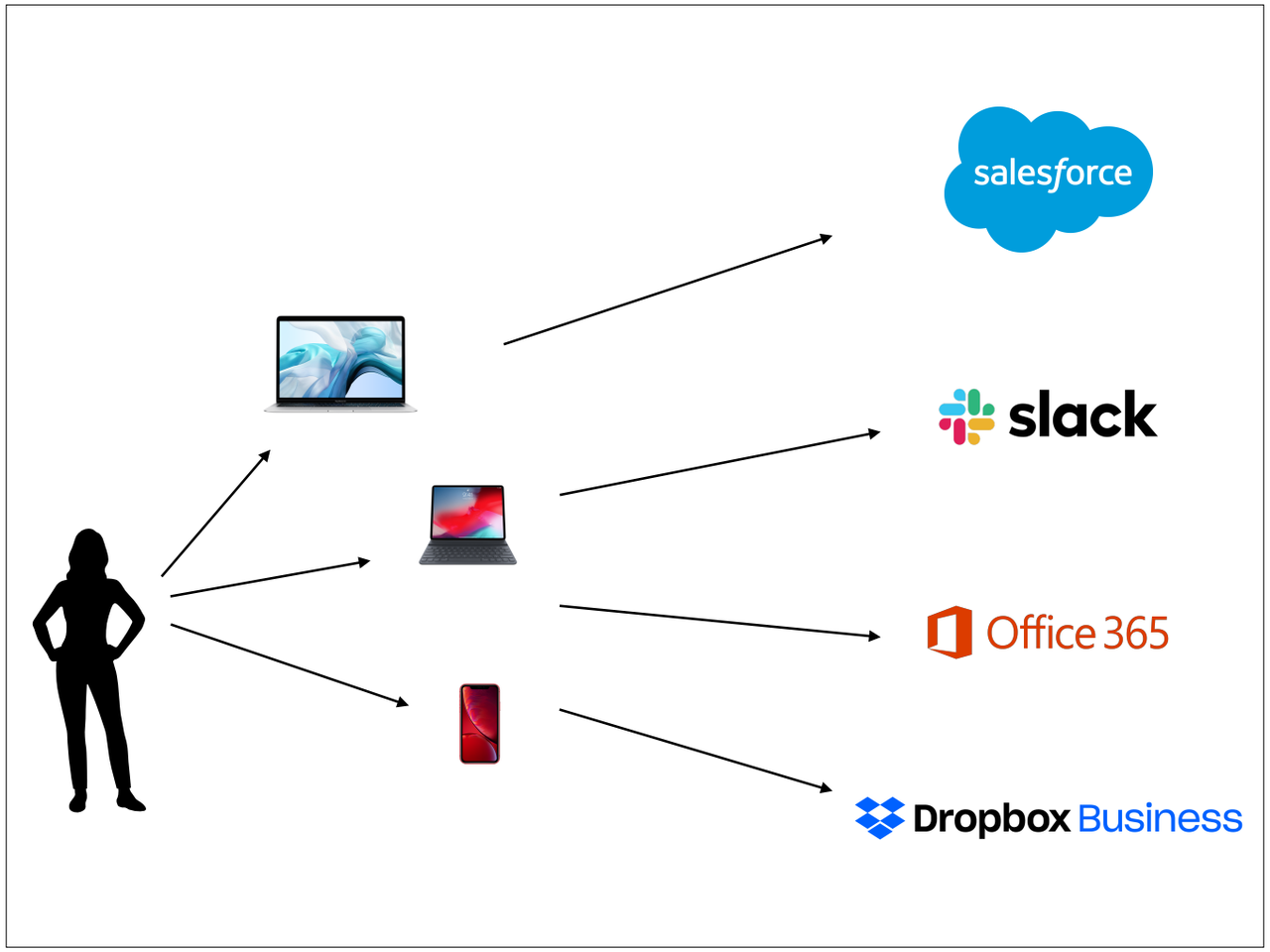
The context

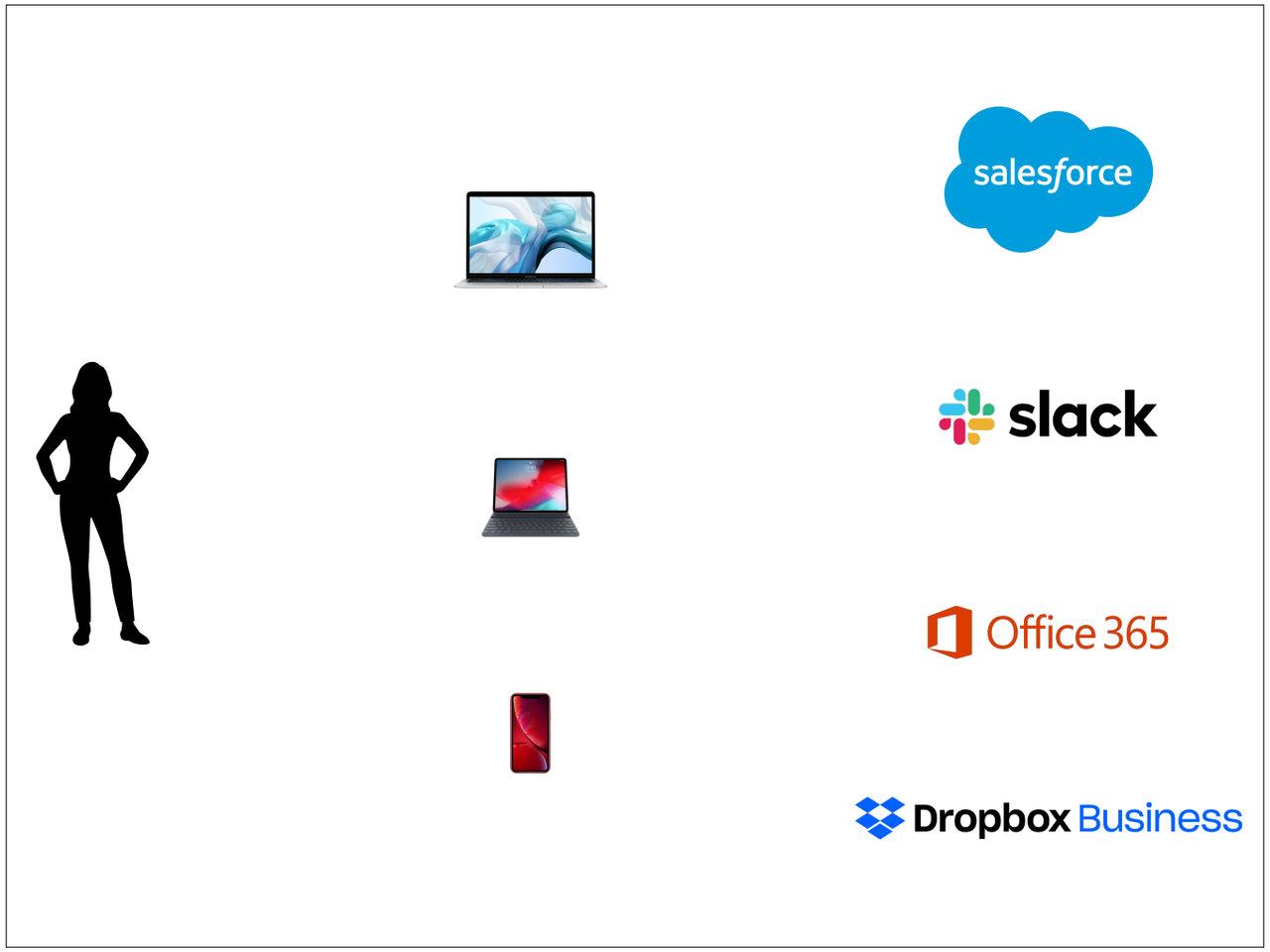
Why your job exist?

Let’s start with the context: why do you exist? What’s the point of your job?

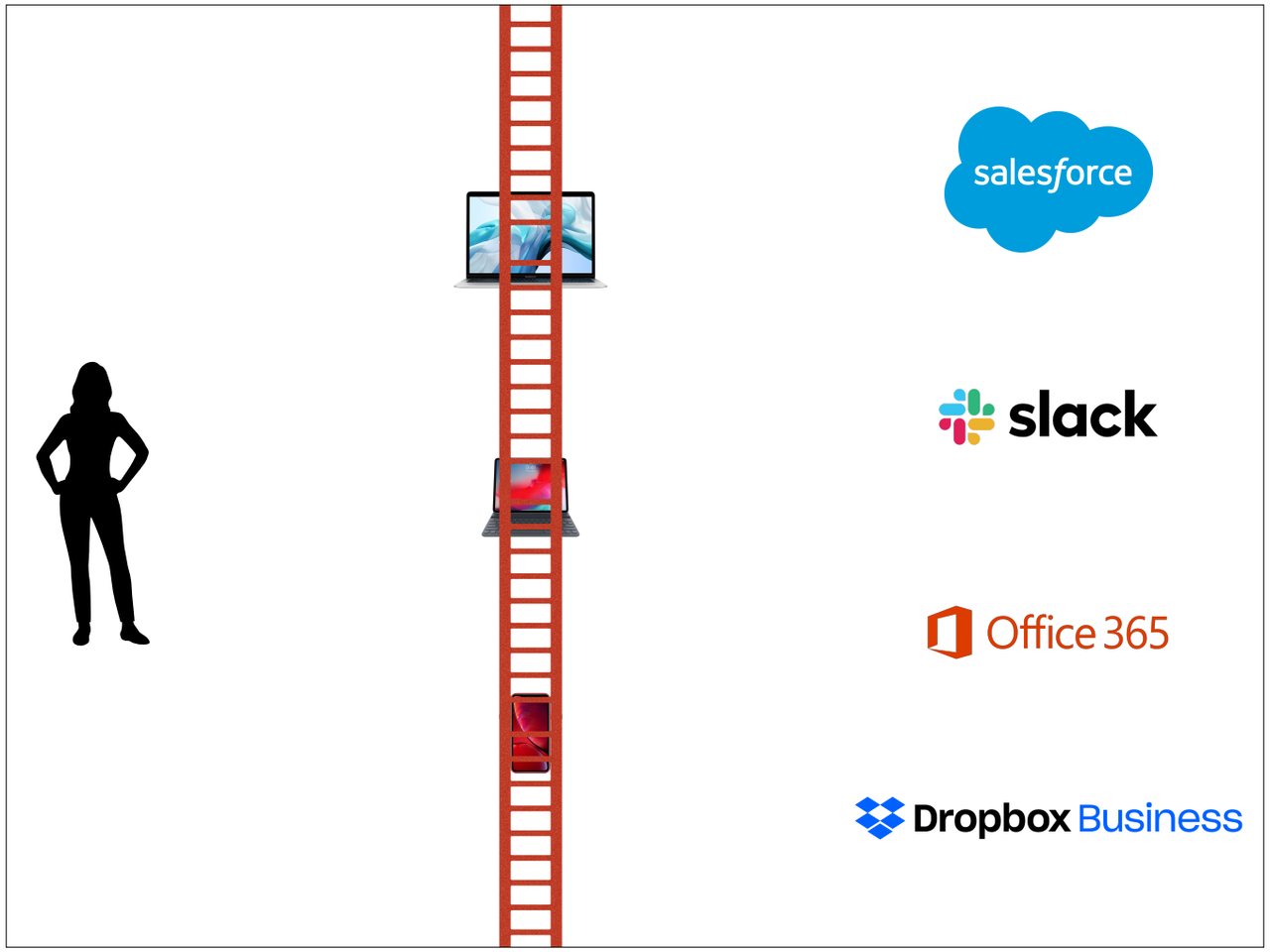
Here is the thing, you company is made of people using devices to access the services their needs



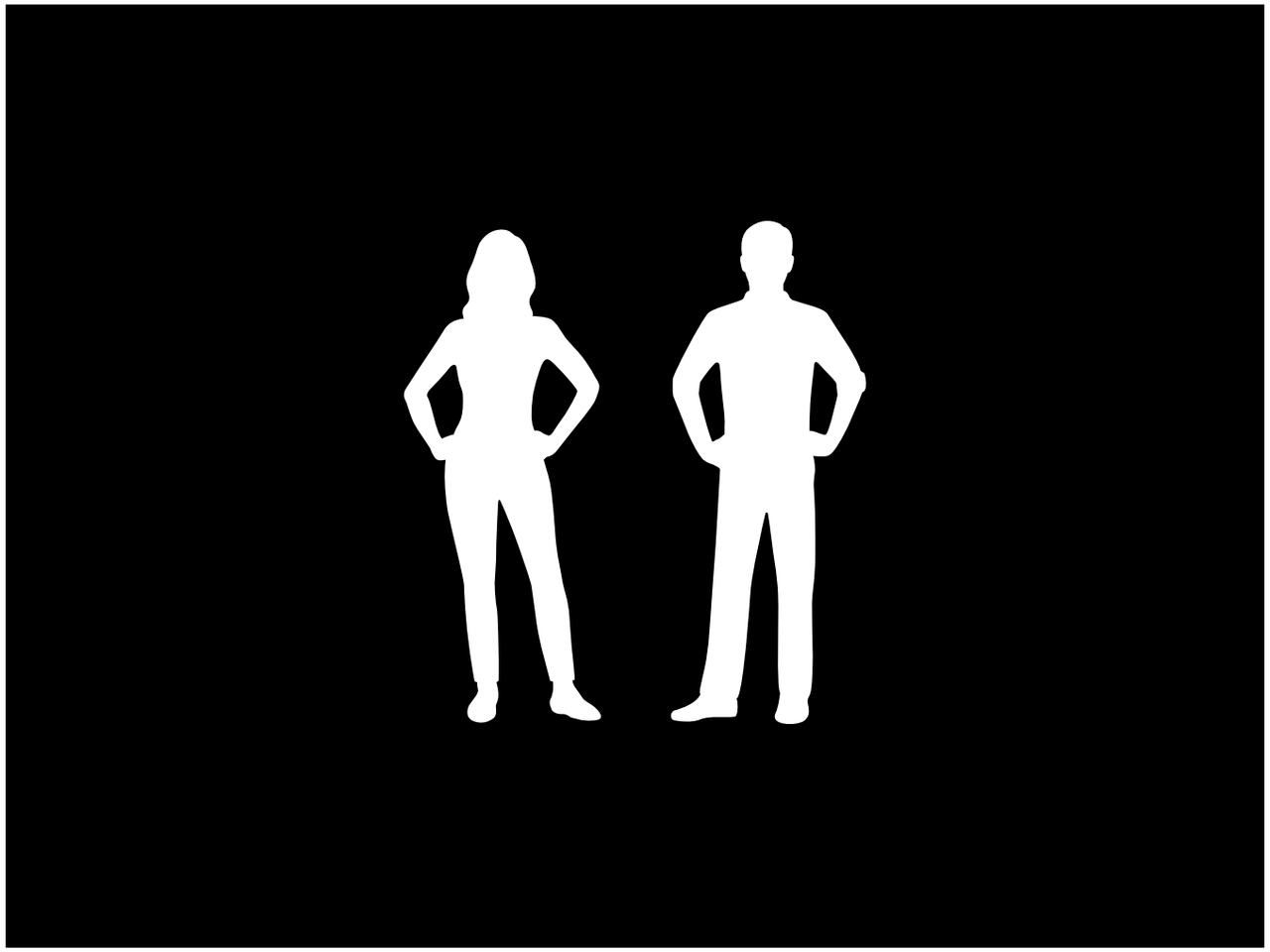

salesforce
slack
Office 365
Dropbox Business

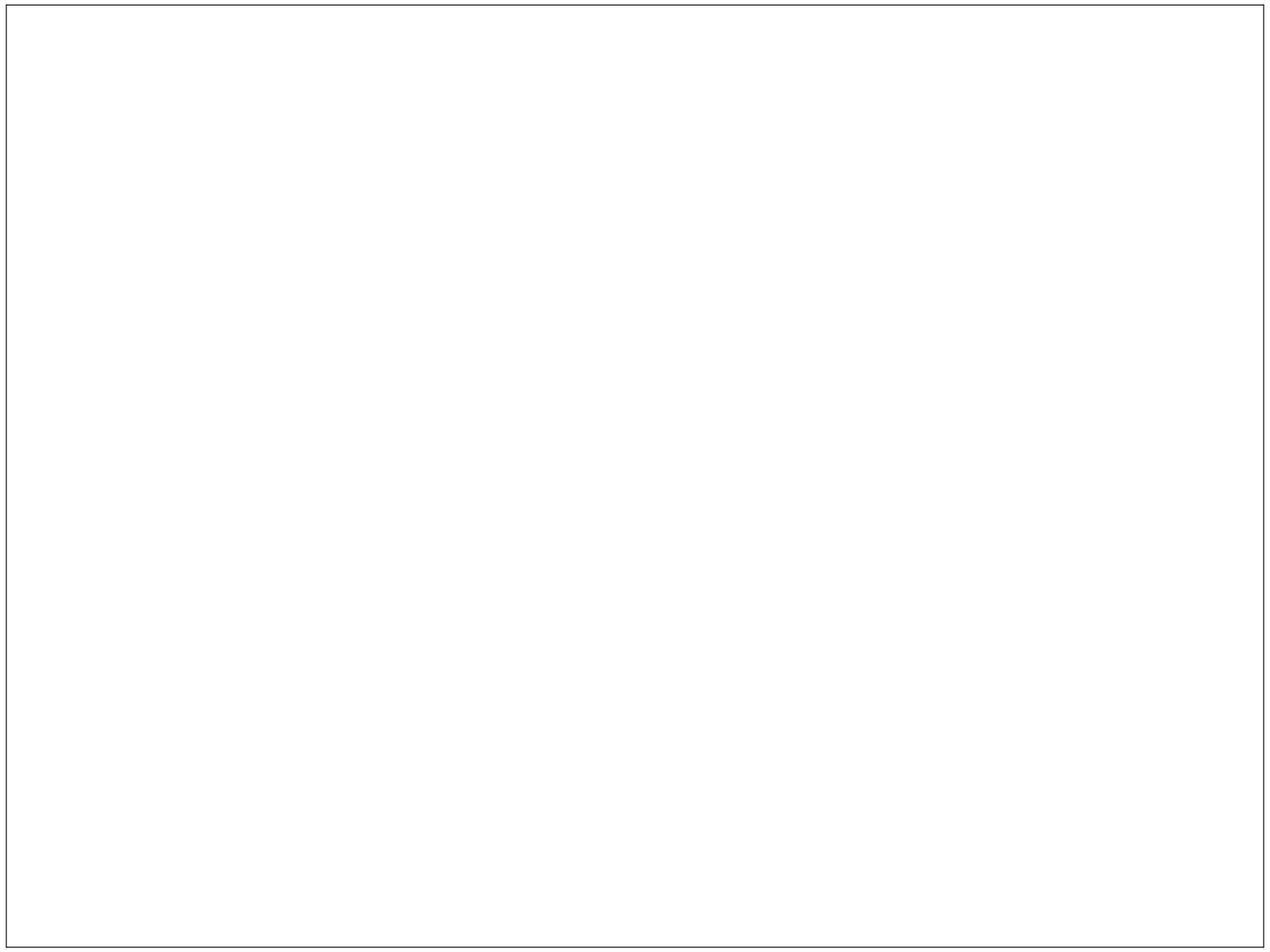


But when you really look at it, for most users, the devices are more acting like a barrier limiting their ability to use what they need to get their job done


salesforce
slack
Office 365
Dropbox Business

Regardless of your organization style, there is always humans, regardless of what they do and who they are, no companies can exist without them.





And your organization need to use services to deliver their own services or products.

RingCentral

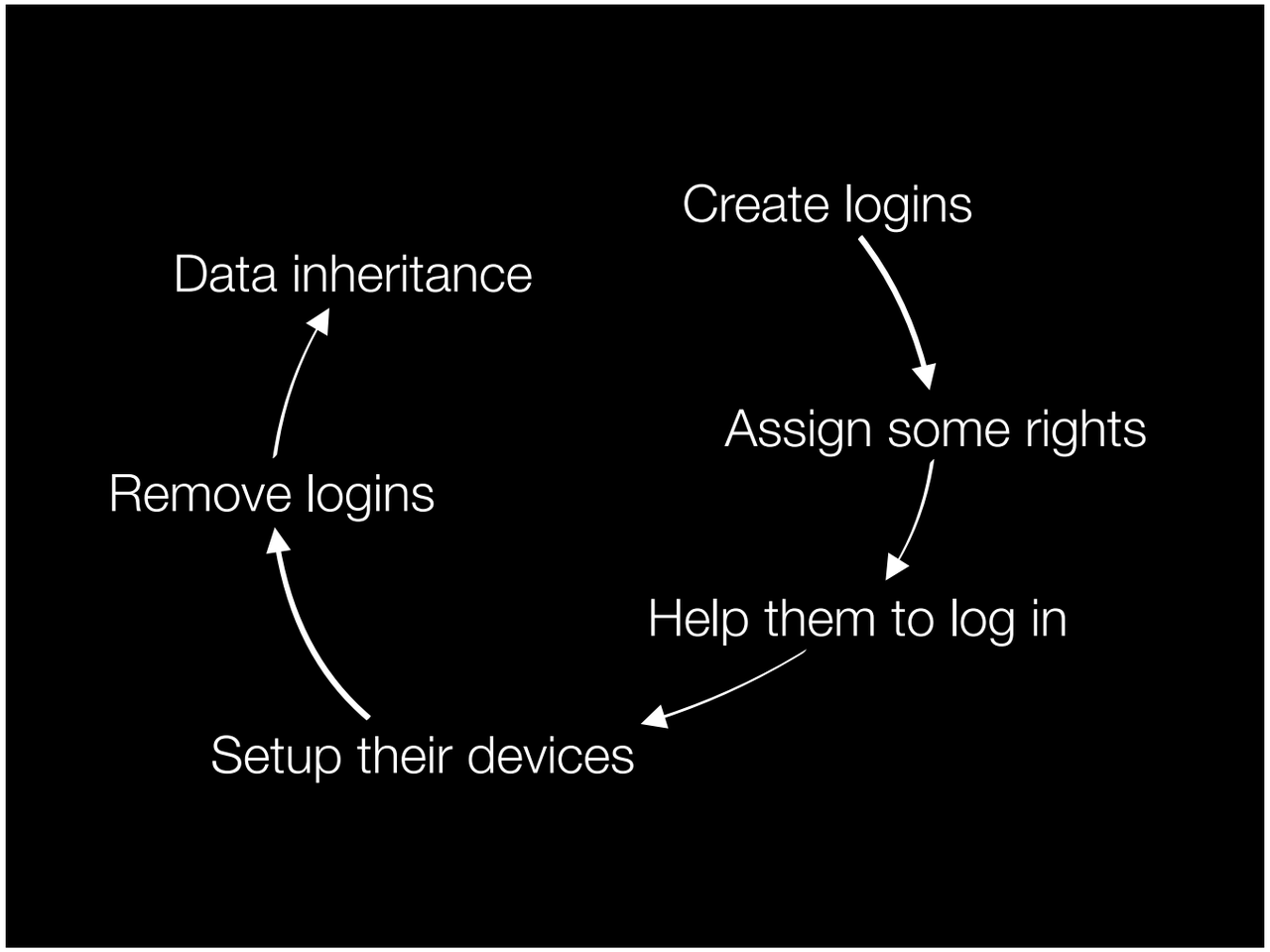
slack

salesforce

Office 365

Dropbox Business

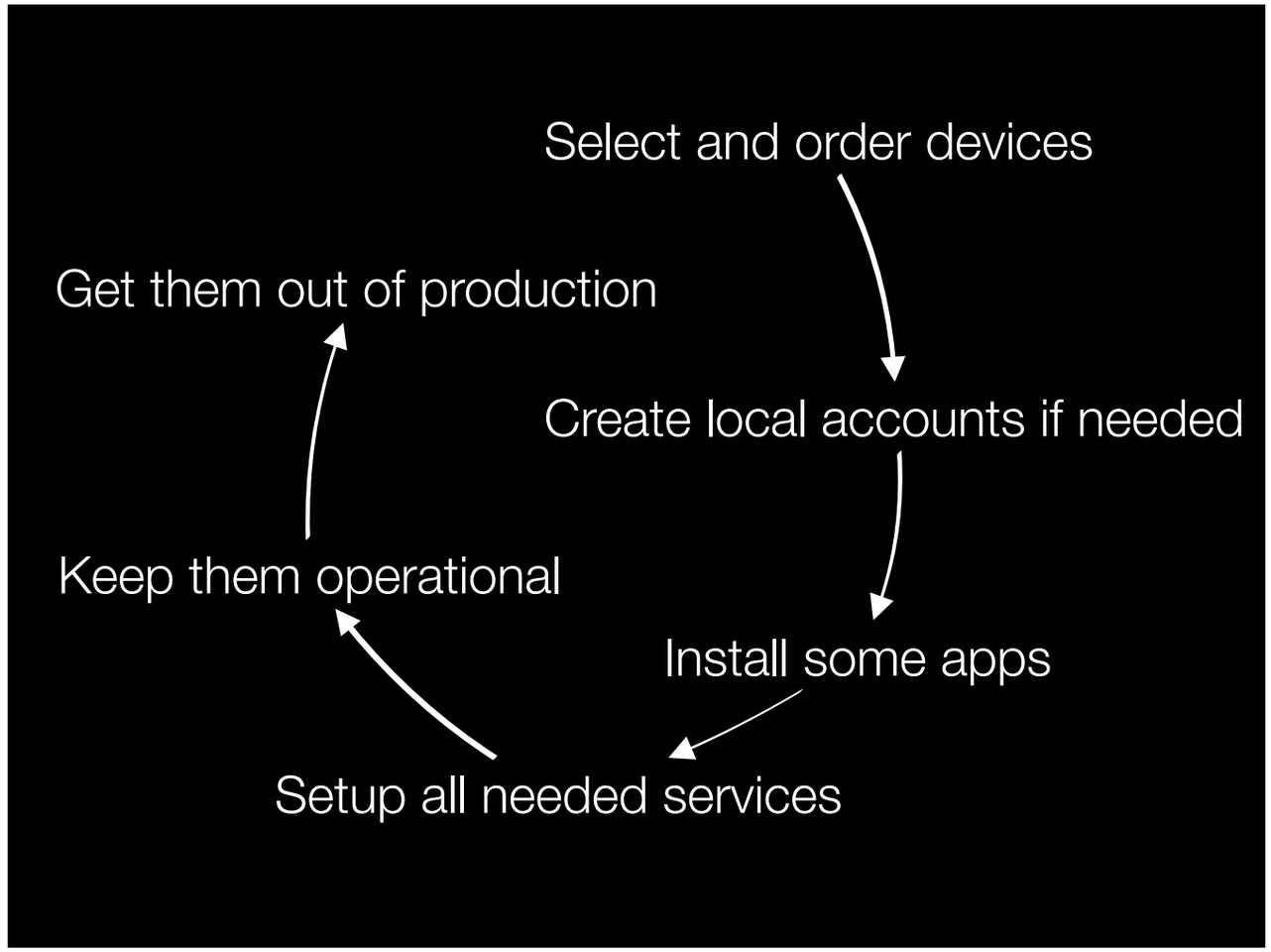
Adobe® Creative Cloud™



So as an IT you will have to handle all identity needs around those services, so creating IDs, managing rights, etc.

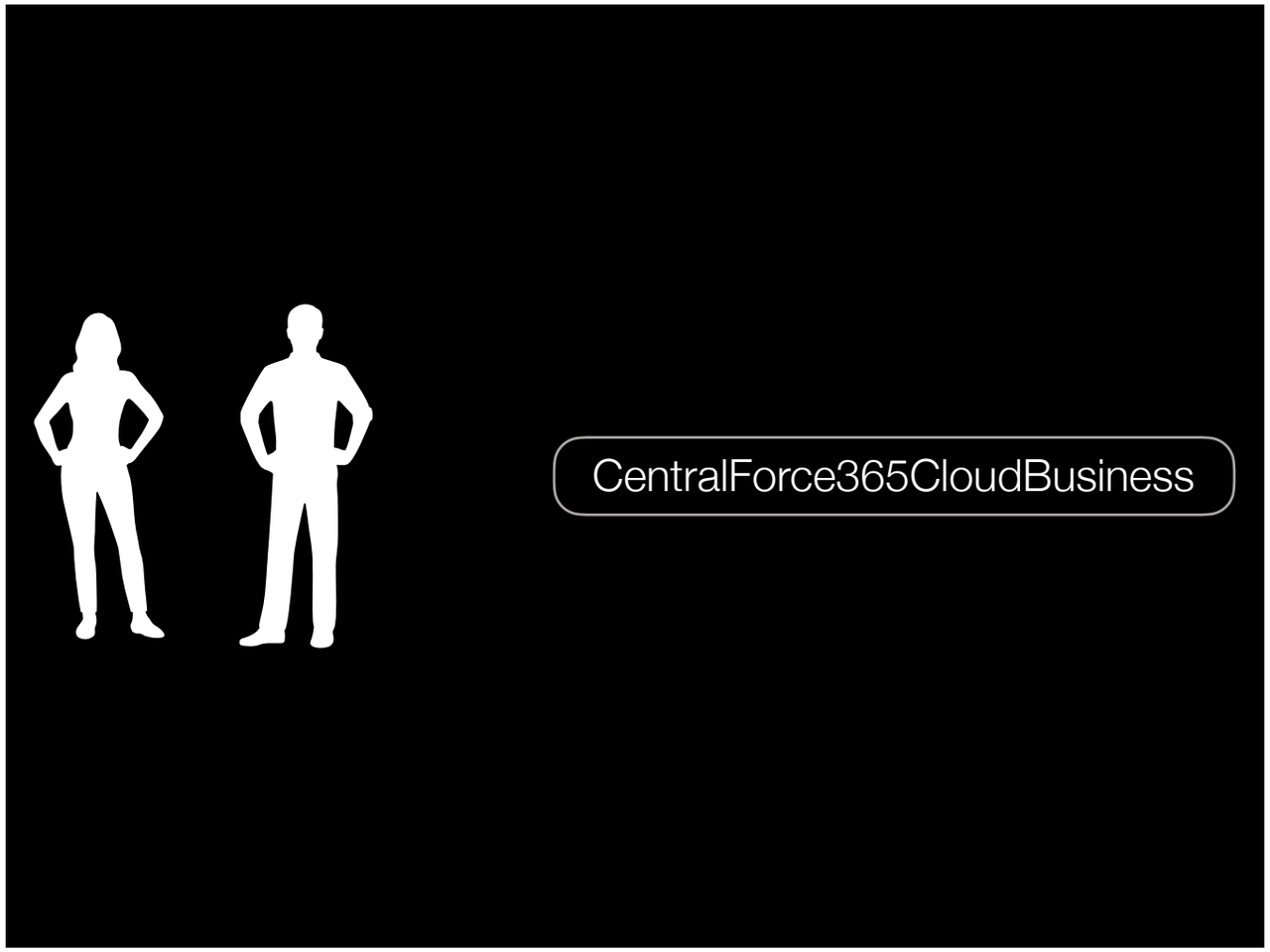
And so, as a requirement to access those services, you will have to provide devices to the end user. But be sure that most of them don’t care as you care to their devices and just want to access the services in the end


9:41



That mean, you will have to handle the whole device lifecycle with account and apps management for them, and you will be in charge of their operational state

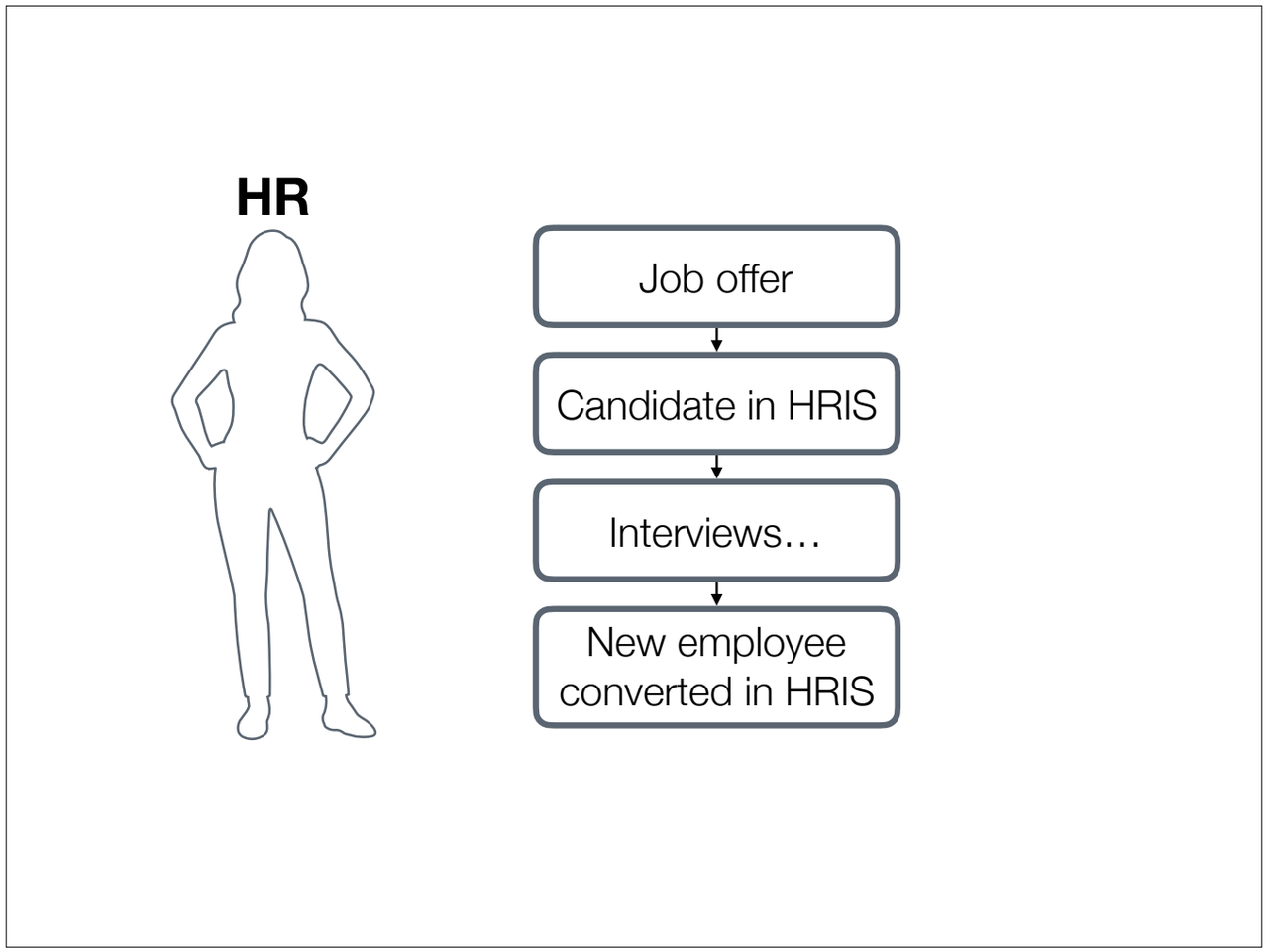
Regardless of your model, never forget that at the end, it’s only users and services. You and your devices are the one who will be responsible for the « I can’t work my computer is broken, it’s IT fault »


CentralForce365CloudBusiness

The workflow

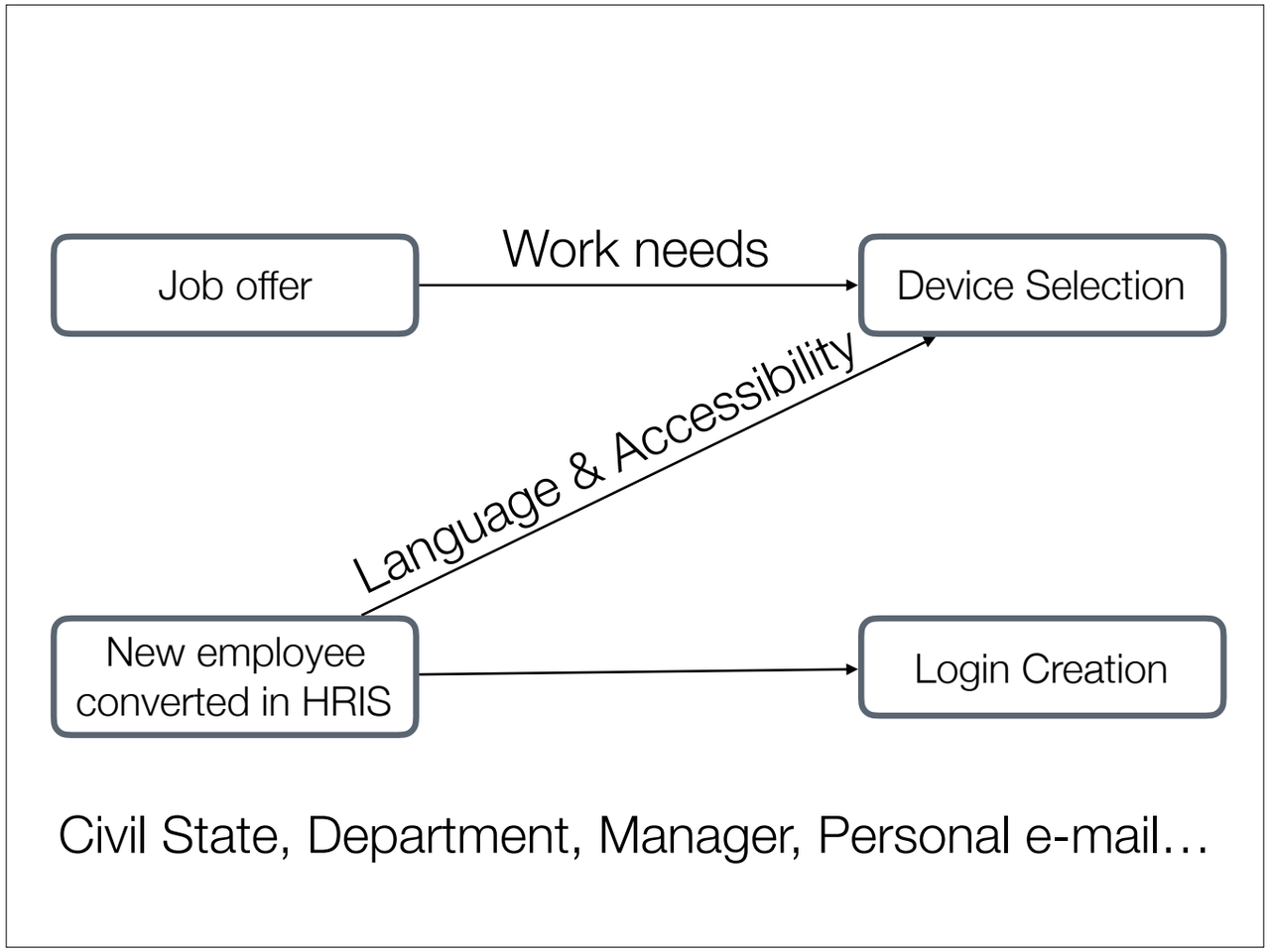
Identity management at its finest

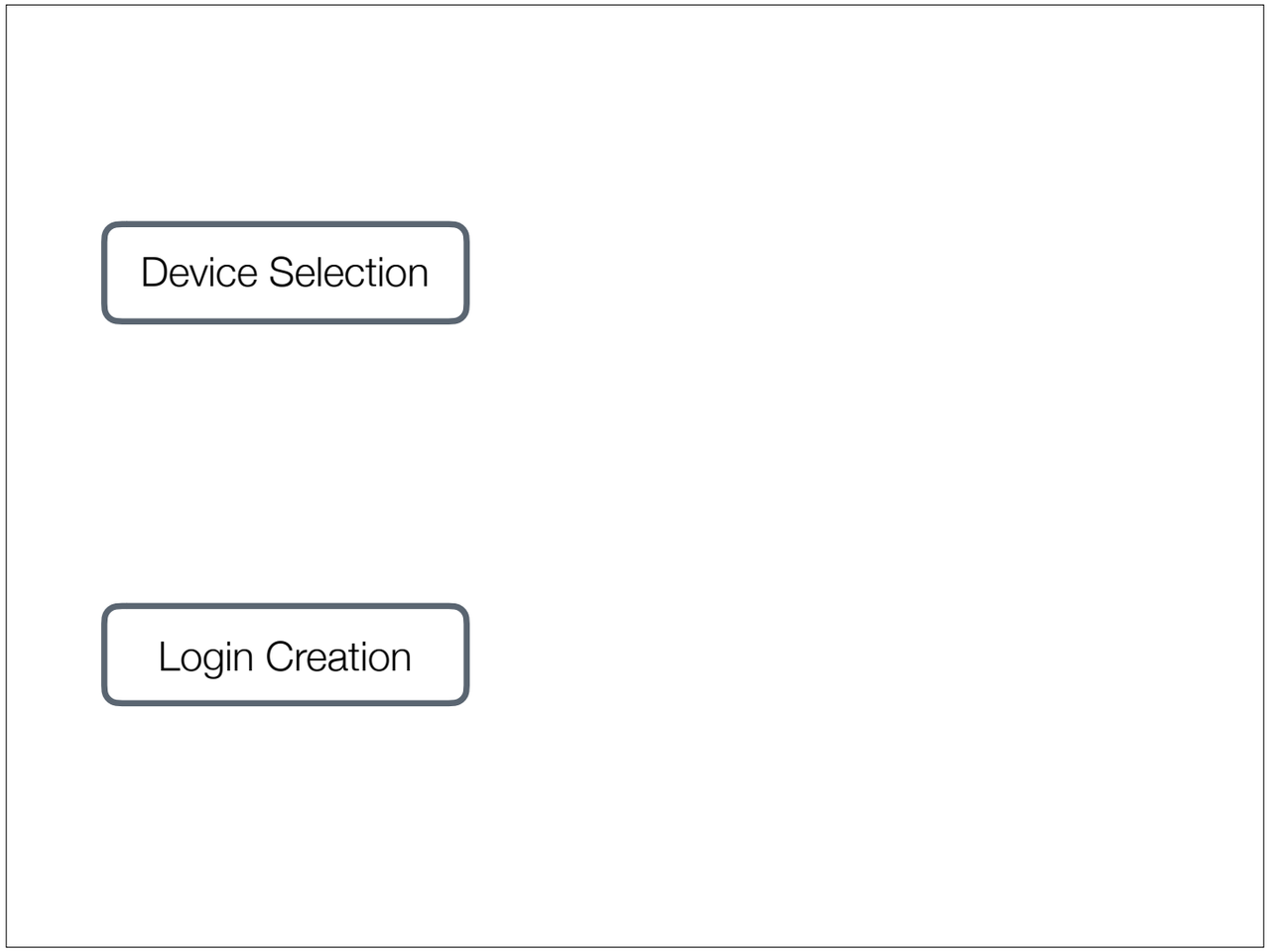
So, let’s think about this identity things, because their is identity everywhere!



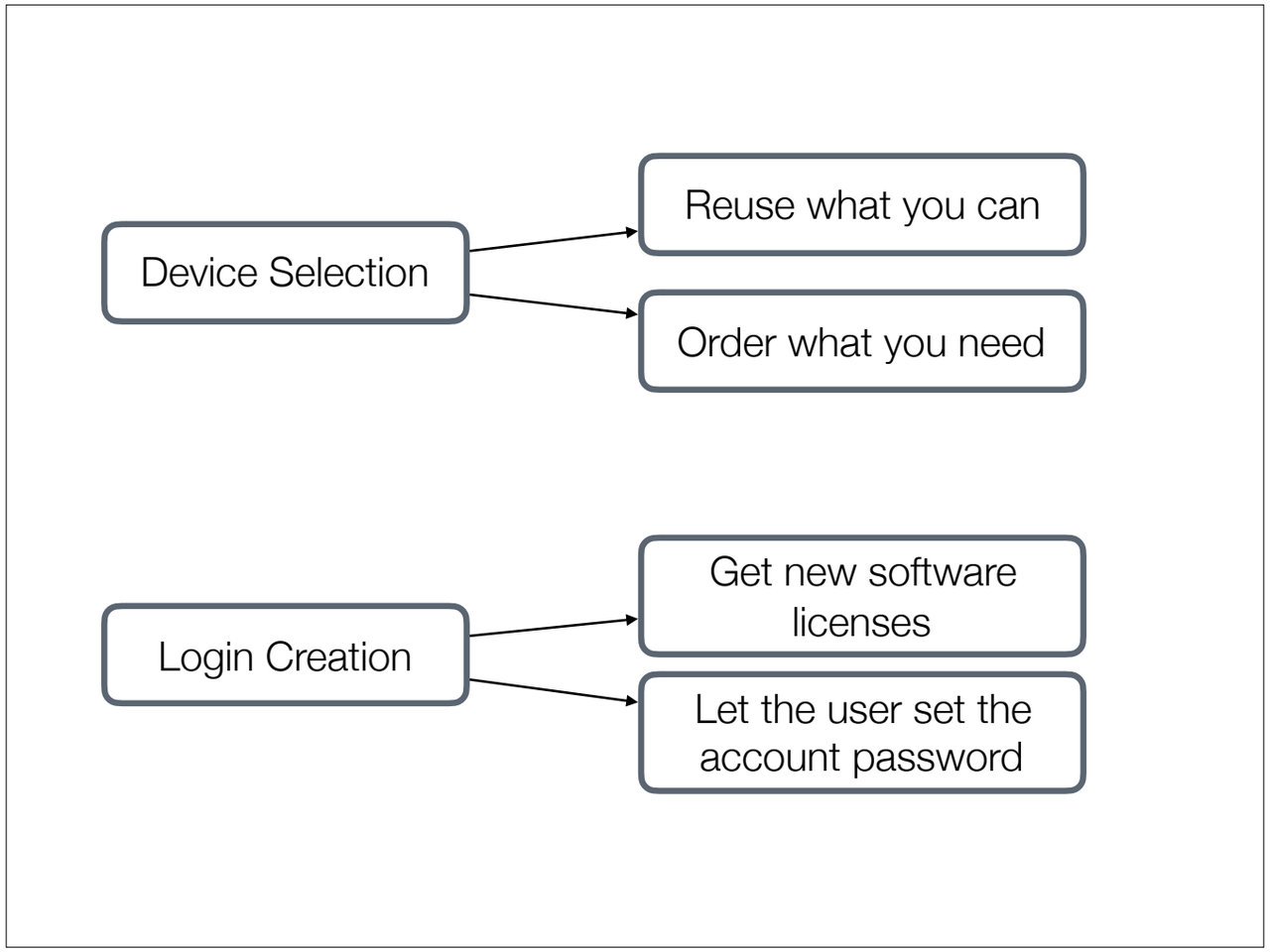
Identity at a company start with Human Resources, with a job offer being open, with candidates being documented in the Human Resources Information System, with interviews and finally with a validated candidate converted in HRIS from candidate state to employee state.

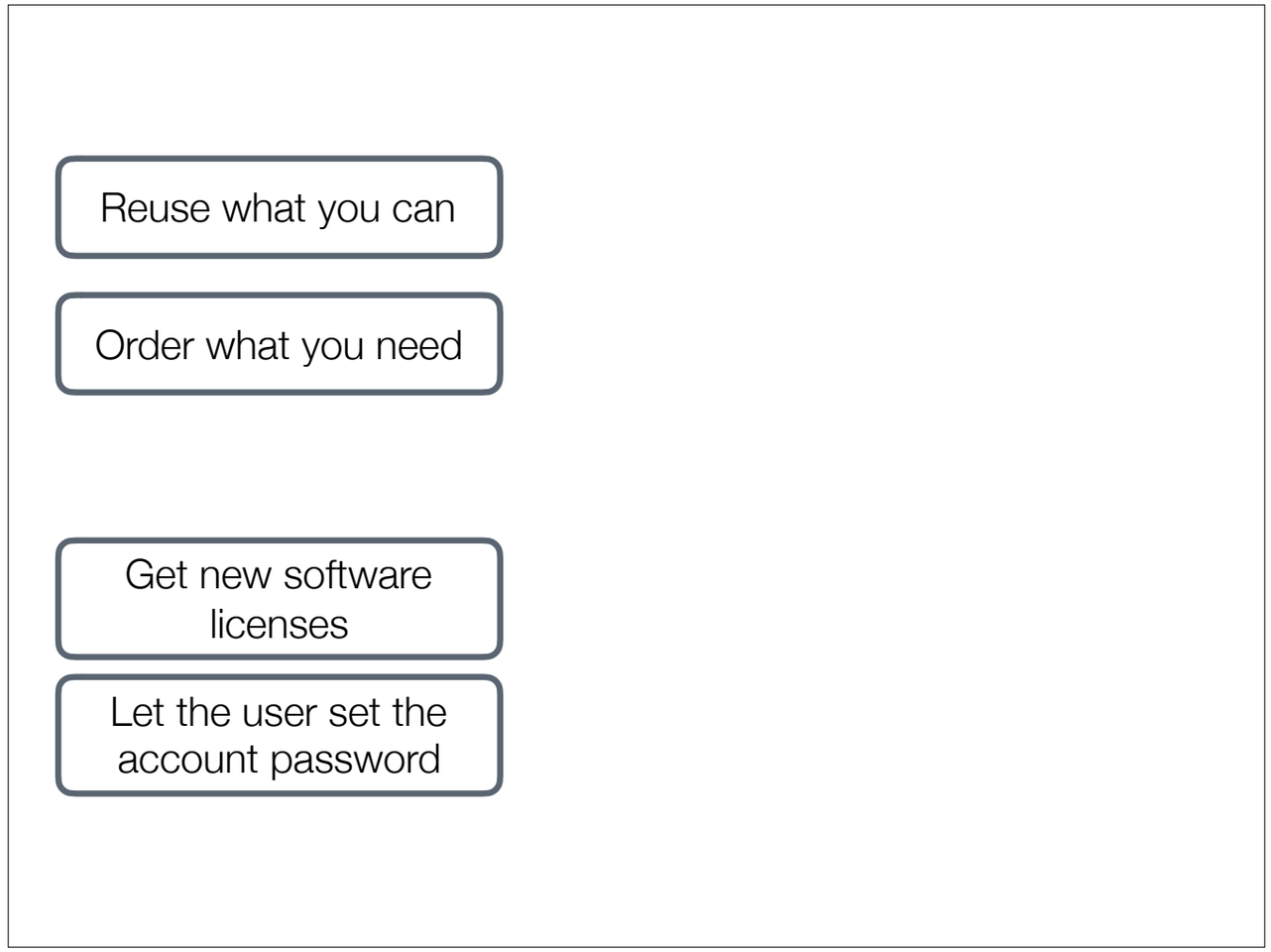
And with those 2 informations, the job offer and the new employee info, you can derivate the device you needs and will obtain all informations needed for the account creation.


Job offer
Work needs
Device Selection
Language & Accessibility
New employee converted in HRIS
Login Creation
Civil State, Department, Manager, Personal e-mail…

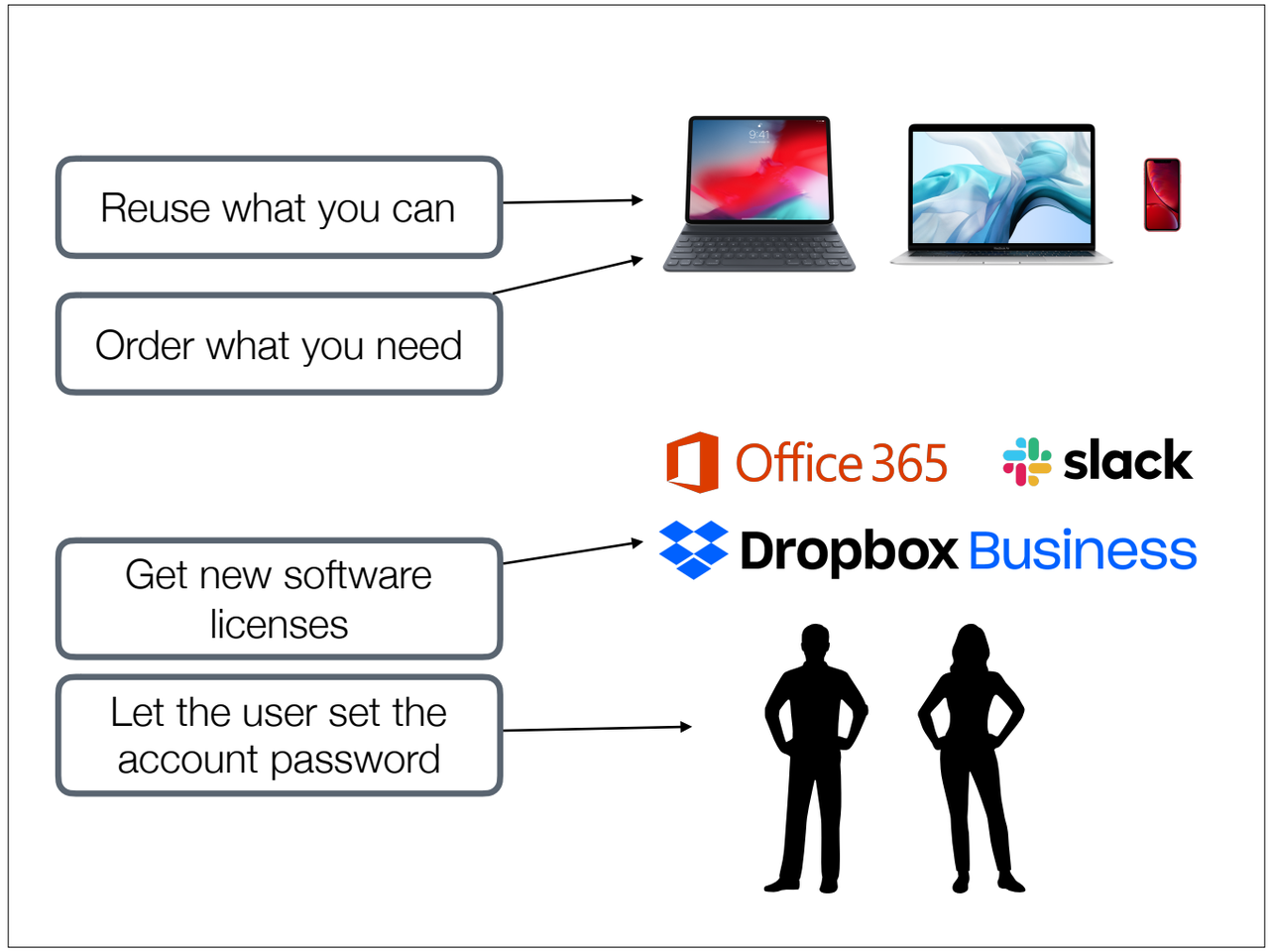


Base on the device selection made you will know if you can reuse existing hardware or if you need new one.
And based on the login info and creation, you will be able to know the software licenses needed and you will eventually be able to provide an on-boarding link to the new employee directly to the private e-mail address so when the person start, ID and password are already known.


Device Selection
Reuse what you can
Order what you need
Login Creation
Get new software licenses
Let the user set the account password



And from that point, you will have everything you need: devices, services and identity.


Reuse what you can
Order what you need
Office 365
slack
Dropbox Business
Get new software licenses
Let the user set the account password

HRIS (and SIS)

Human Resources Informations System as well as Student Informations System are the best way for you as IT to get trustable informations about your users. By directly interfacing your directory service with those system you can automate everything needed.

HRIS (and SIS)

Provide trustable info related to users

HRIS (and SIS)

Provide trustable info related to users

Must be accessible via API or automated CSV export

HRIS (and SIS)

Provide trustable info related to users

Must be accessible via API or automated CSV export

Should also inform about teams for group management

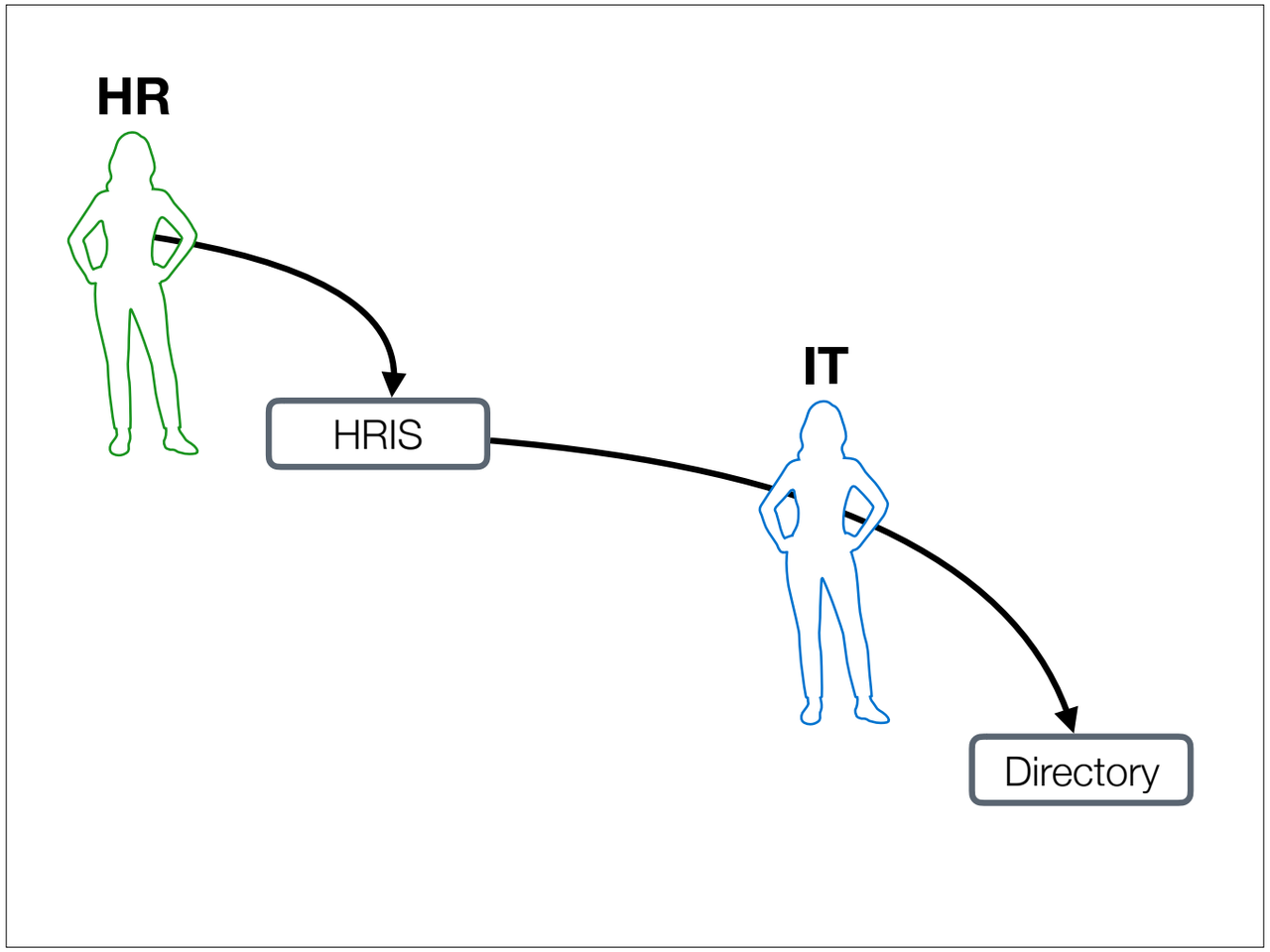
HRIS (and SIS)

Provide trustable info related to users

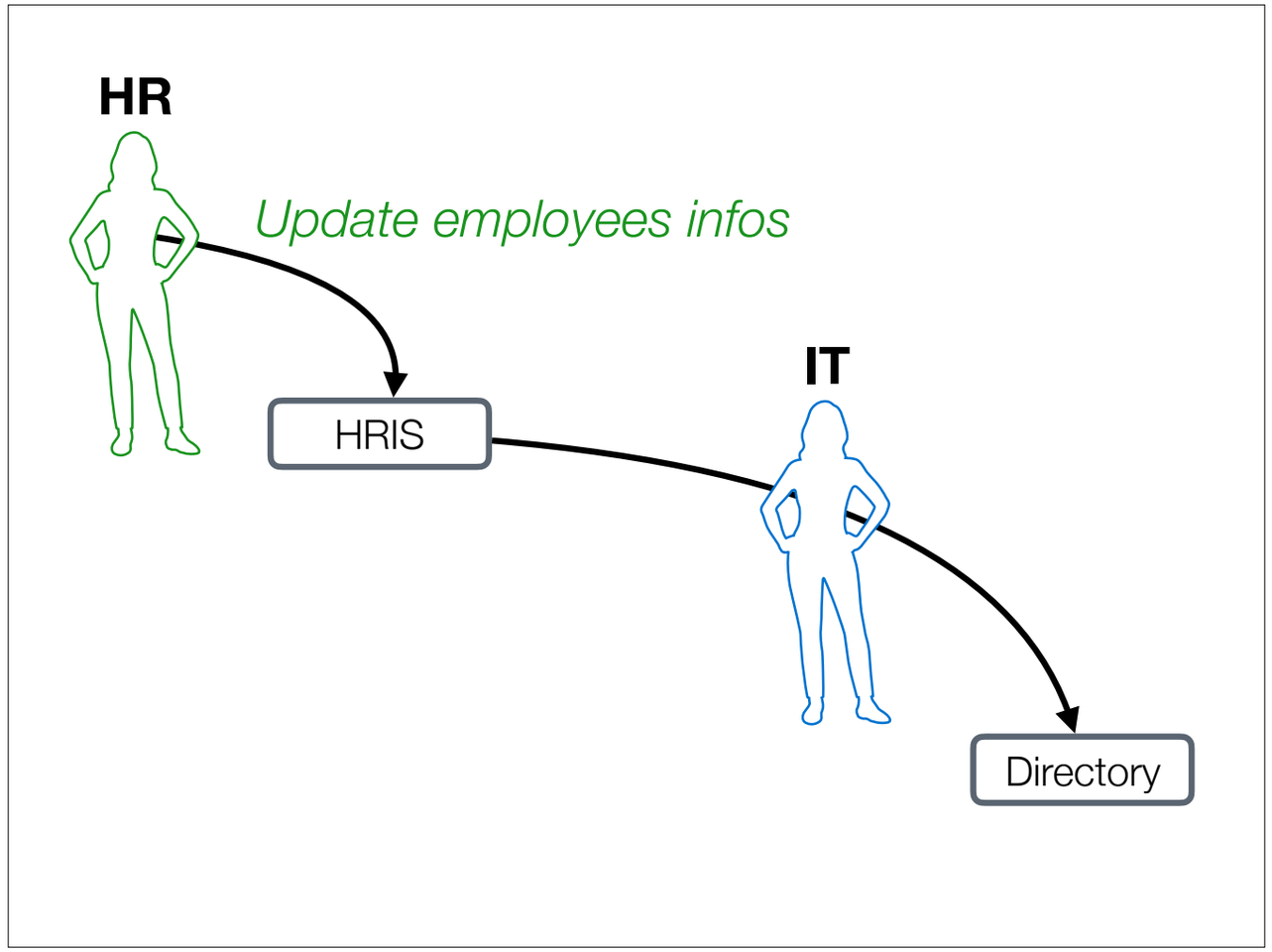
Must be accessible via API or automated CSV export

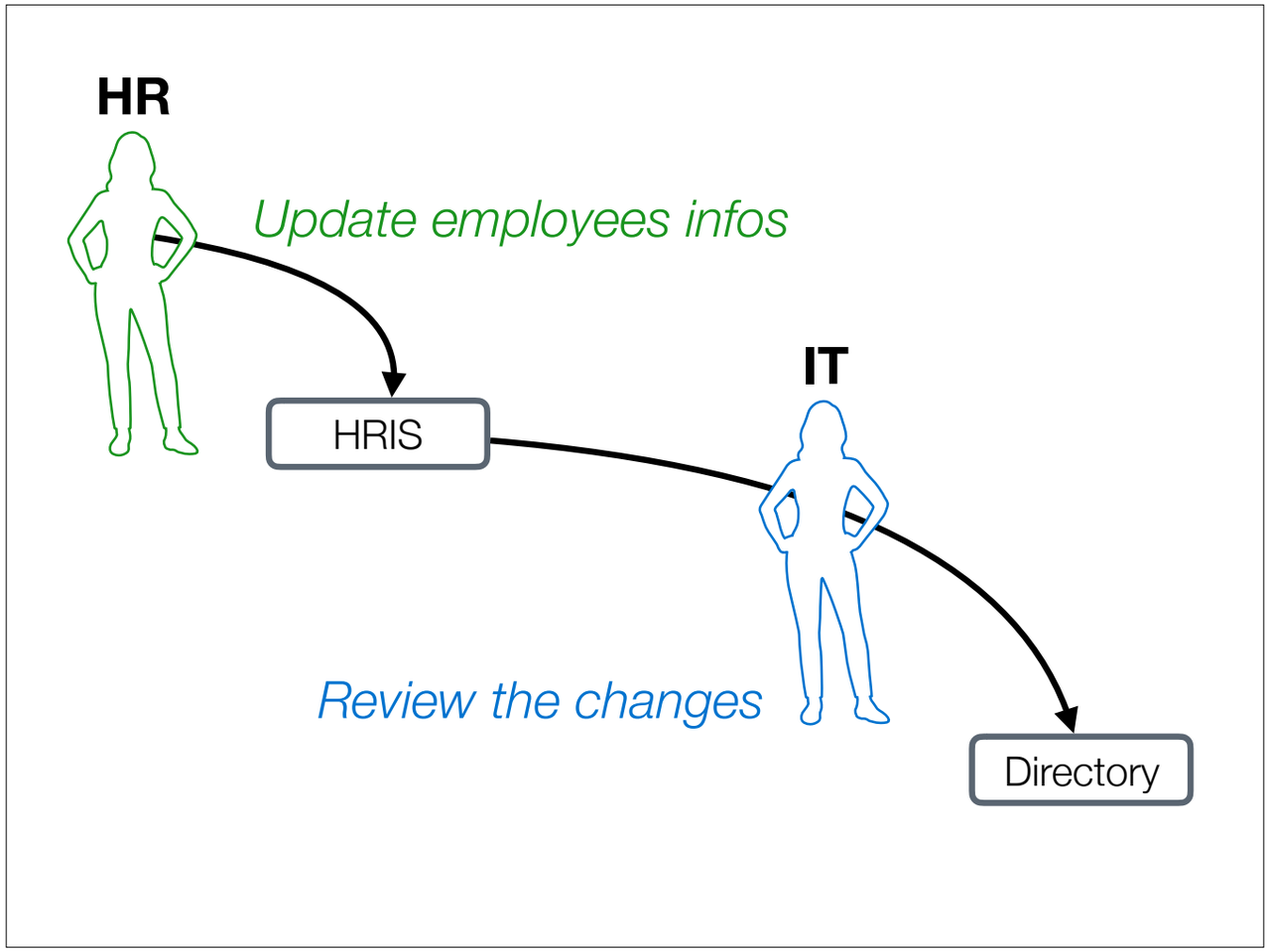
Should also inform about teams for group management

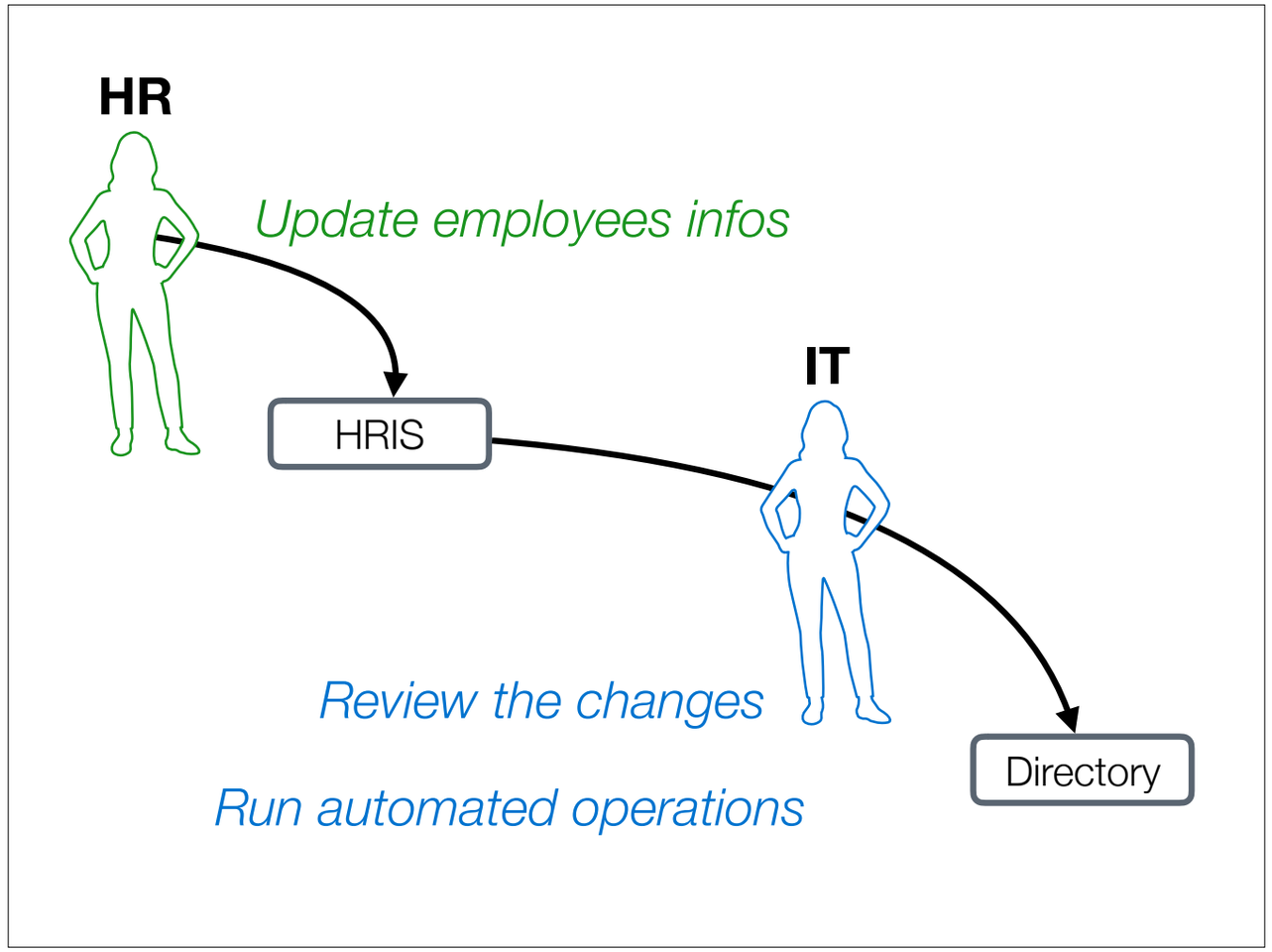
Native integration / custom script to your directory



Main idea is here, HR people fill HRIS with employee info, and you, running automation tool will simply have to check that HR work is done well (no typo or whatever) and run script to do the work, avoiding so human error in between the HR info and the directory


HR
Update employees infos
HRIS
IT
Directory

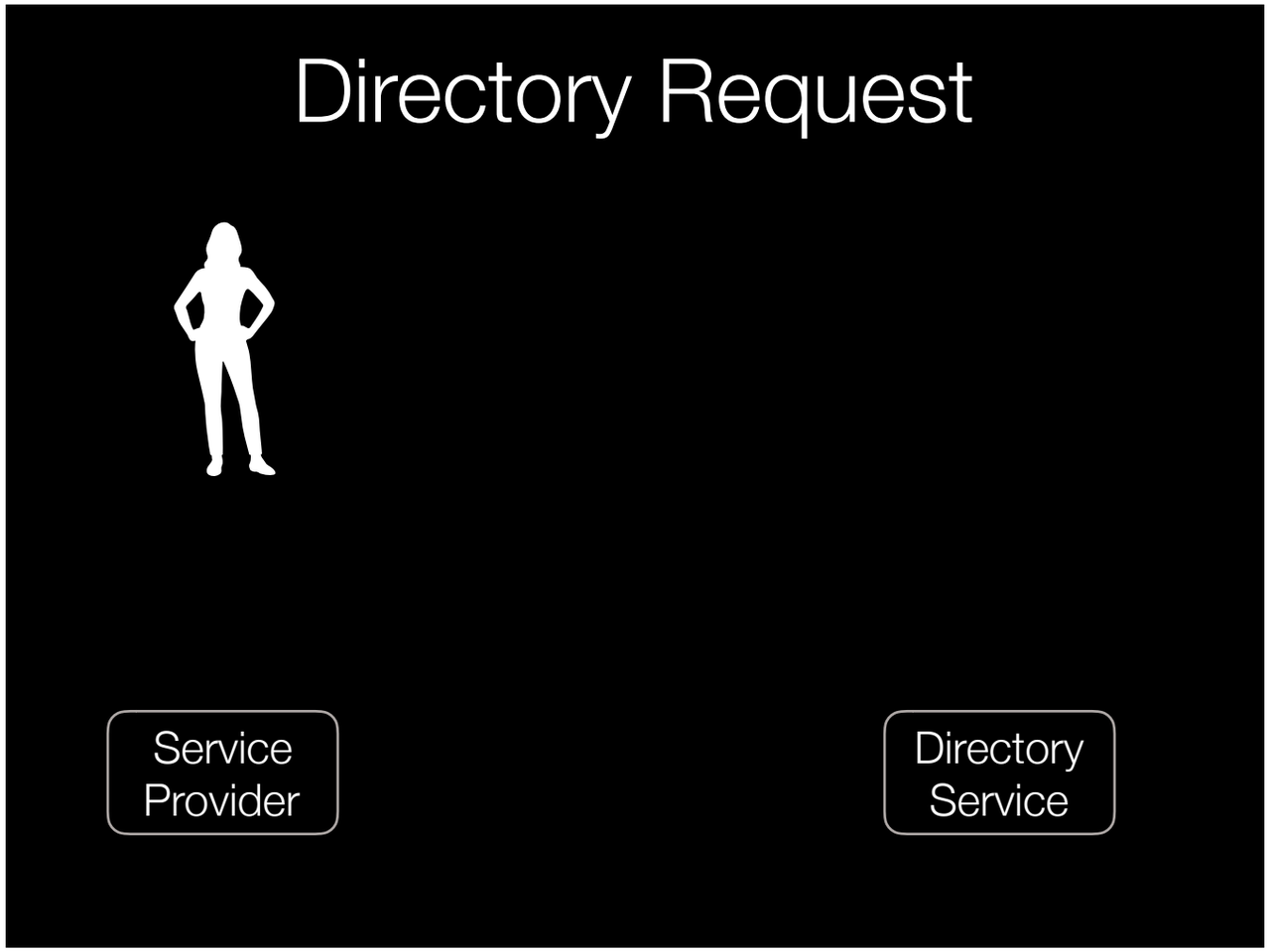

HR
Update employees infos
HRIS
IT
Review the changes
Directory


HR
Update employees infos
HRIS
IT
Review the changes
Directory
Run automated operations

Accessing the identities

User provisioning or directory request?

Now that we have identities, we need our services to access it, and their is many way to do that



The usual model that you all know, is the directory request model used by LDAP. When the user ask something to a Service Provider that require user info, a directory request is sent by the Service Provider to the Directory Service, and so the Directory Service answer the request and then the Service Provider can do its job.
Doing so mean that, if you Service Provider is cloud based and your Directory Service on premise, behind a firewall, you will need to open access at firewall level.
In such scenario, the Service Provider MUST reach the Directory Service to get things done!

Directory Request

Lookup Request

Service Provider

Directory Service

Directory Request

Lookup Request

Directory Request

Service Provider

Directory Service

Directory Request

Lookup Request

Directory Request

Directory Request Result

Service Provider

Directory Service

Directory Request

Lookup Request

Directory Request

Directory Request Result

Lookup Request Result

Service Provider

Directory Service

Directory Request

Lookup Request

Directory Request

Directory Request Result

Lookup Request Result

Service Provider

Directory Service

Service Provider must reach the Directory Service

Directory Request

Lookup Request

Directory Request

Directory Request Result

Lookup Request Result

Service Provider

Directory Service

Service Provider must reach the Directory Service

Directory Request

Lookup Request

Directory Request

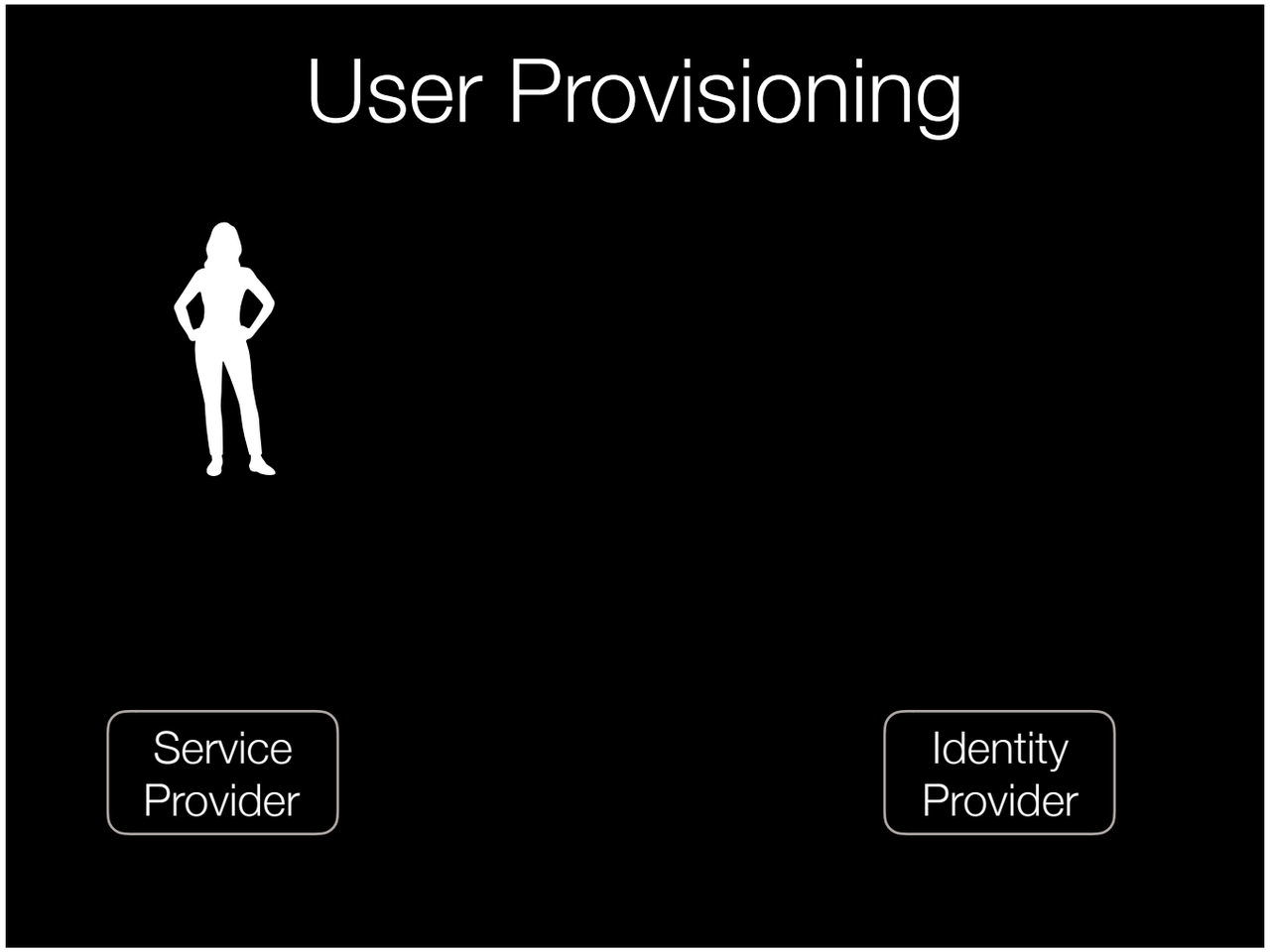
Directory Request Result

Lookup Request Result

Service Provider

Directory Service

Service Provider must reach the Directory Service



The other way of doing this is with the User Provisioning model, where the Identity Provider will now be in charge to push changes to all services before it need it, so when the user ask for something, the Service Provider is already able to answer.

User Provisioning

Inform for identity change

Service Provider

Identity Provider

User Provisioning

Inform for identity change

Lookup Request

Service Provider

Identity Provider

User Provisioning

Inform for identity change

Lookup Request

Lookup Request Result

Service Provider

Identity Provider

User Provisioning

Inform for identity change

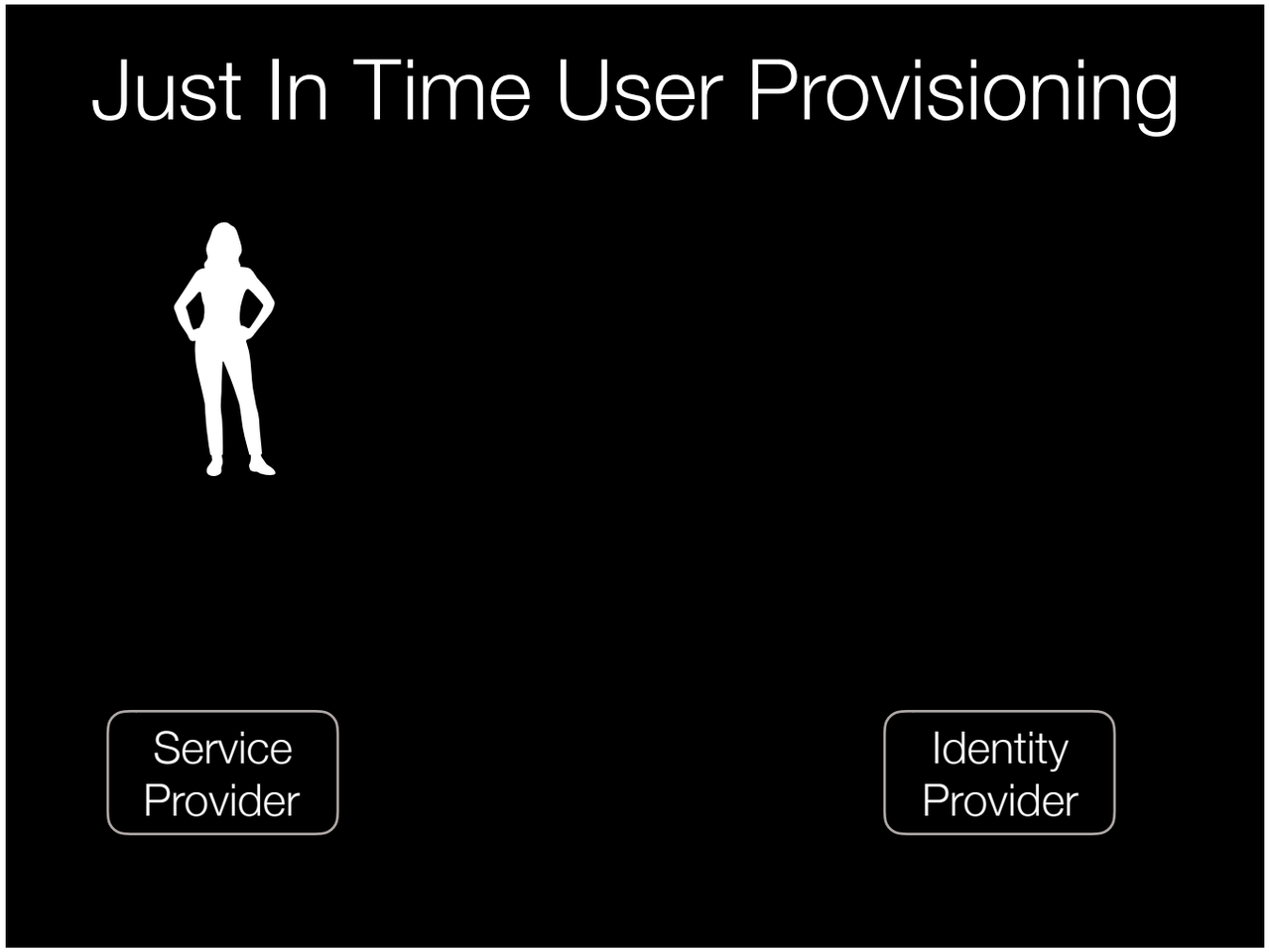
Lookup Request

Lookup Request Result

Service Provider

Identity Provider

Service Provider does not access the Identity Provider



There is also another User Provisioning method called the Just In Time method, it will make sense in few slides but basically, when the Service Provider rely on the Identity Provider for the authentication, it will be able to use the proof of authentication to serve as an identity update.

Just In Time User Provisioning

Authentication Request

Service Provider

Identity Provider

Just In Time User Provisioning

Authentication Request

Authentication Result

Service Provider

Identity Provider

Just In Time User Provisioning

Authentication Request

Authentication Result

Proof of identity

Service Provider

Identity Provider

Just In Time User Provisioning

Authentication Request

Authentication Result

Proof of identity

Service Provider

Identity Provider

Service Provider can update on the fly with proof of identity

Identity Management

Modern authentication and provisioning

Now that we understand the main differences between legacy and modern identity management, let’s dive into the workflow and technics around modern authentication and provisioning.

Modern authentication

Modern authentication can rely on a lot of solution like OpenID, SAML, OAuth, or WS-Fed. The overall scenario is mainly the same with all

Modern authentication

OpenID

SAML

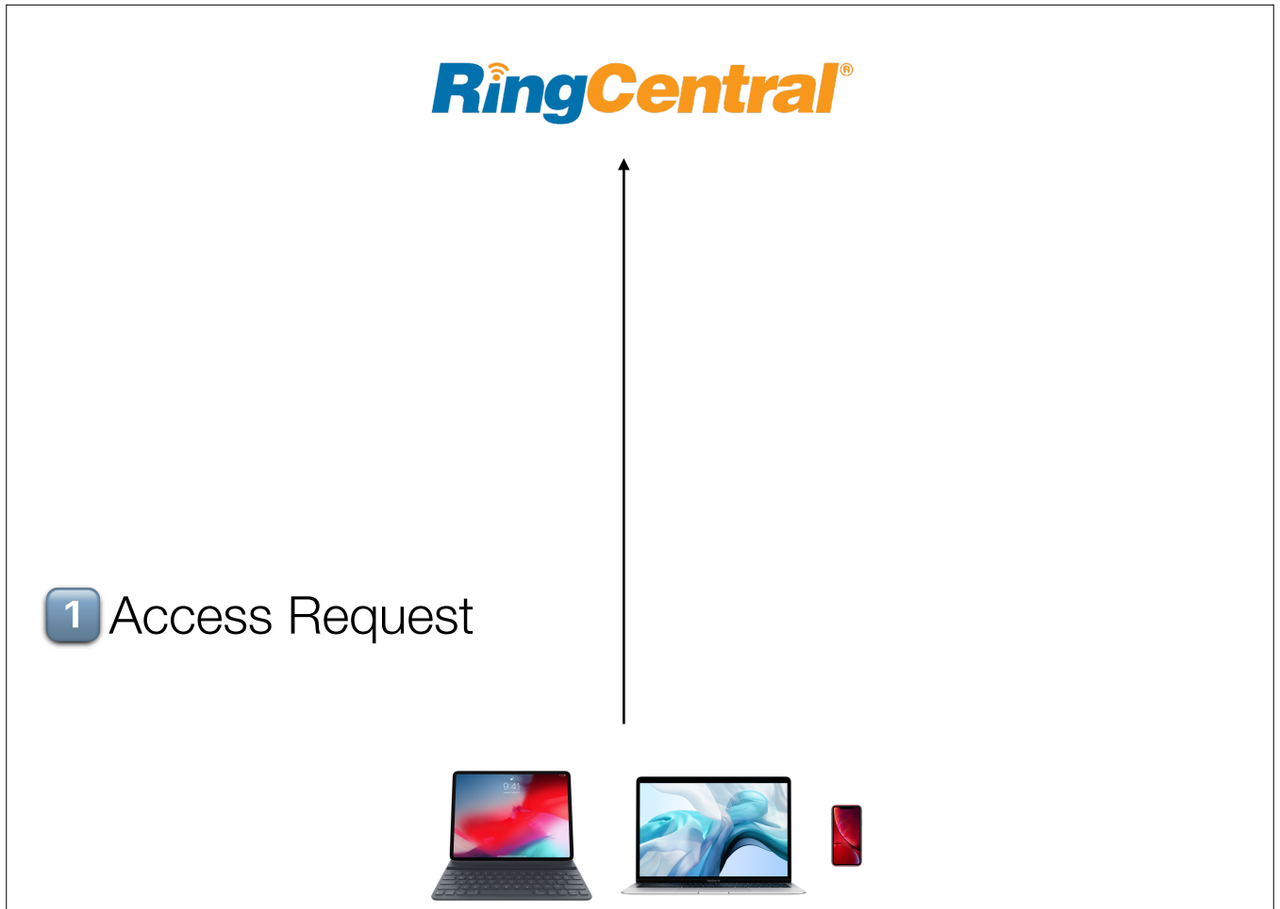
OAuth

WS-Fed

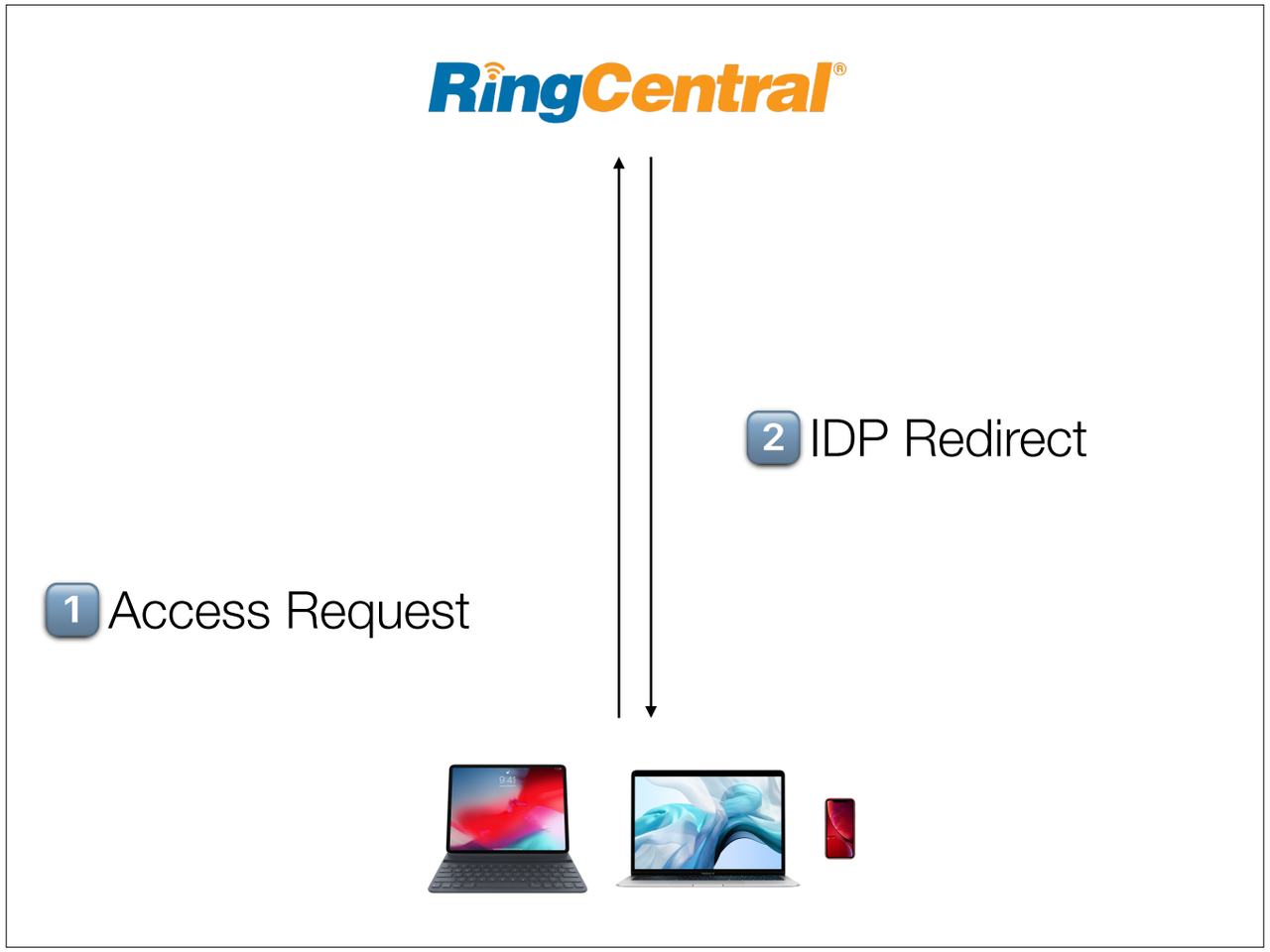
Let say your user want to access their enterprise phone solution, an access request will be sent from the device to the Service Provider

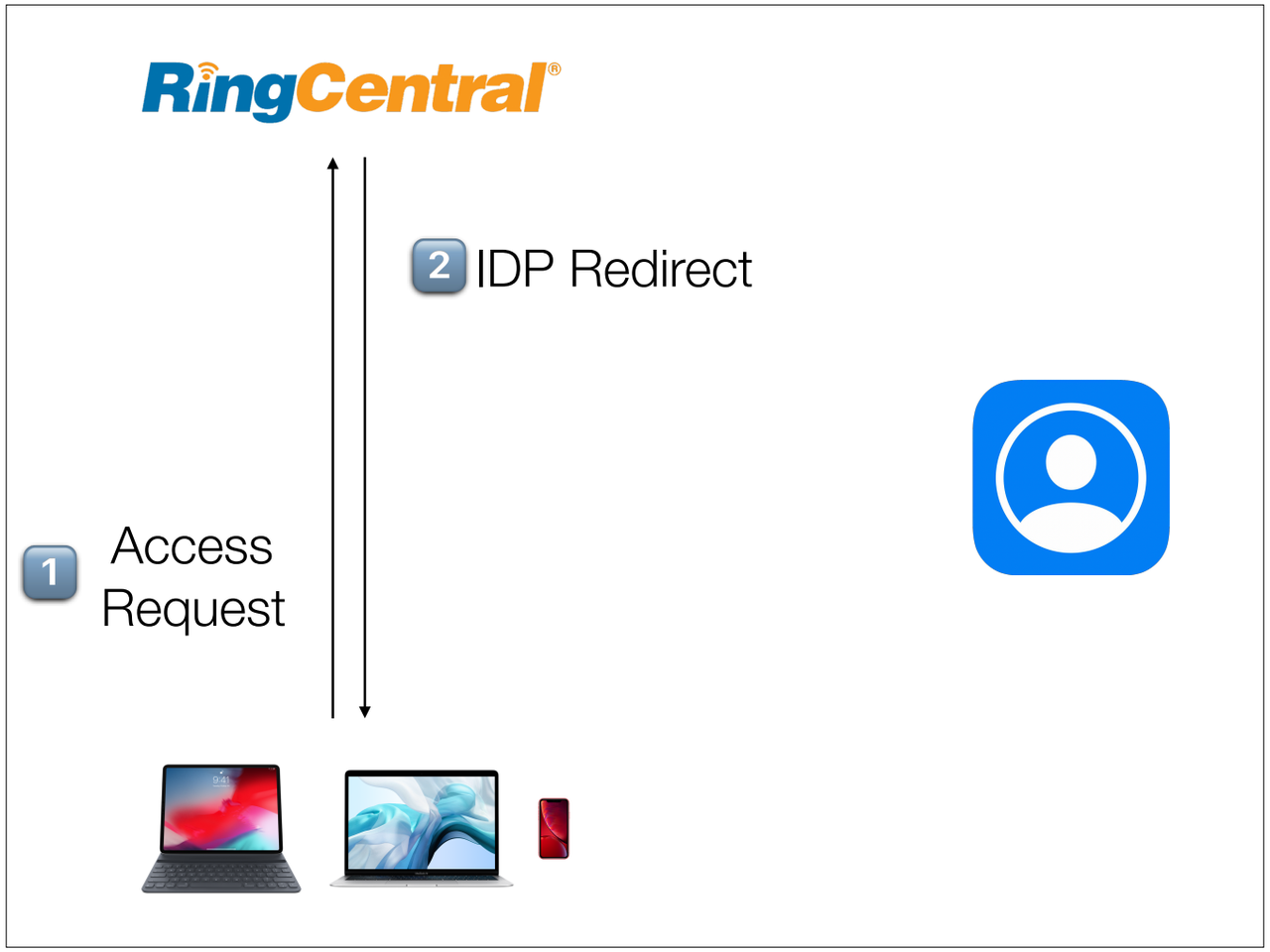
RingCentral

1 Access Request

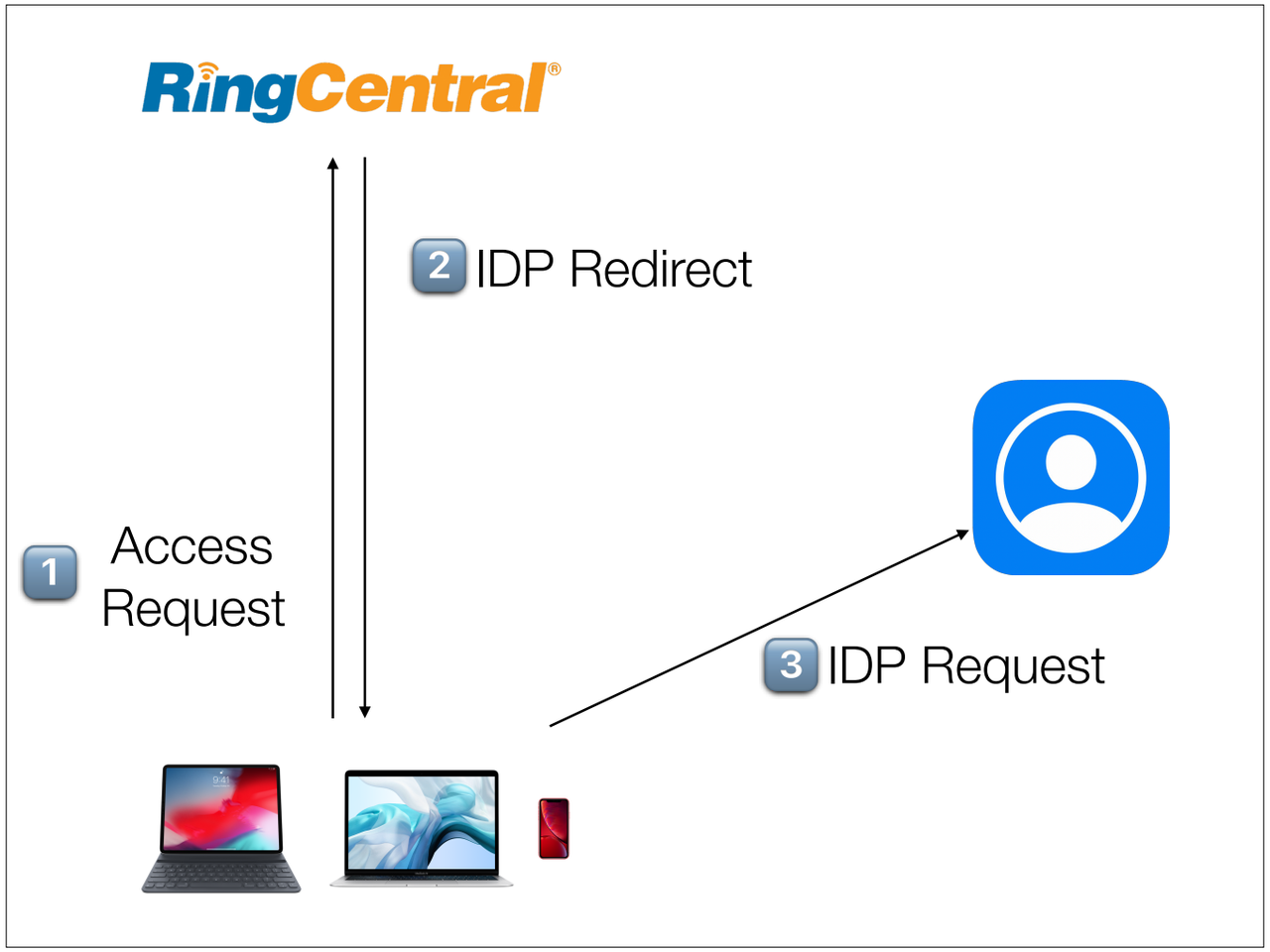


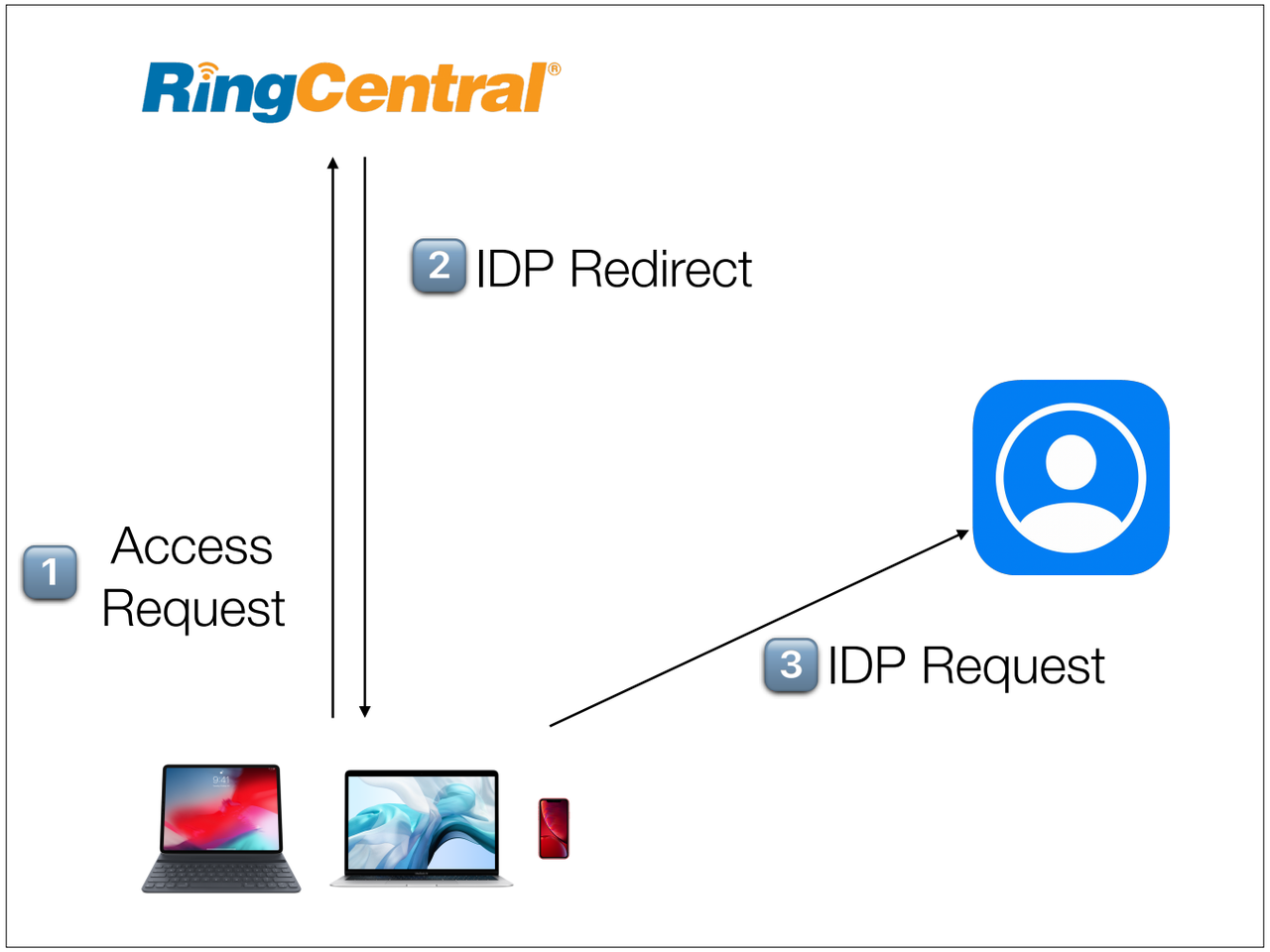
If we are in a modern authentication scenario, the service provider will redirect the user to the Identity Provider for authentication purpose


RingCentral
1 Access Request
2 IDP Redirect

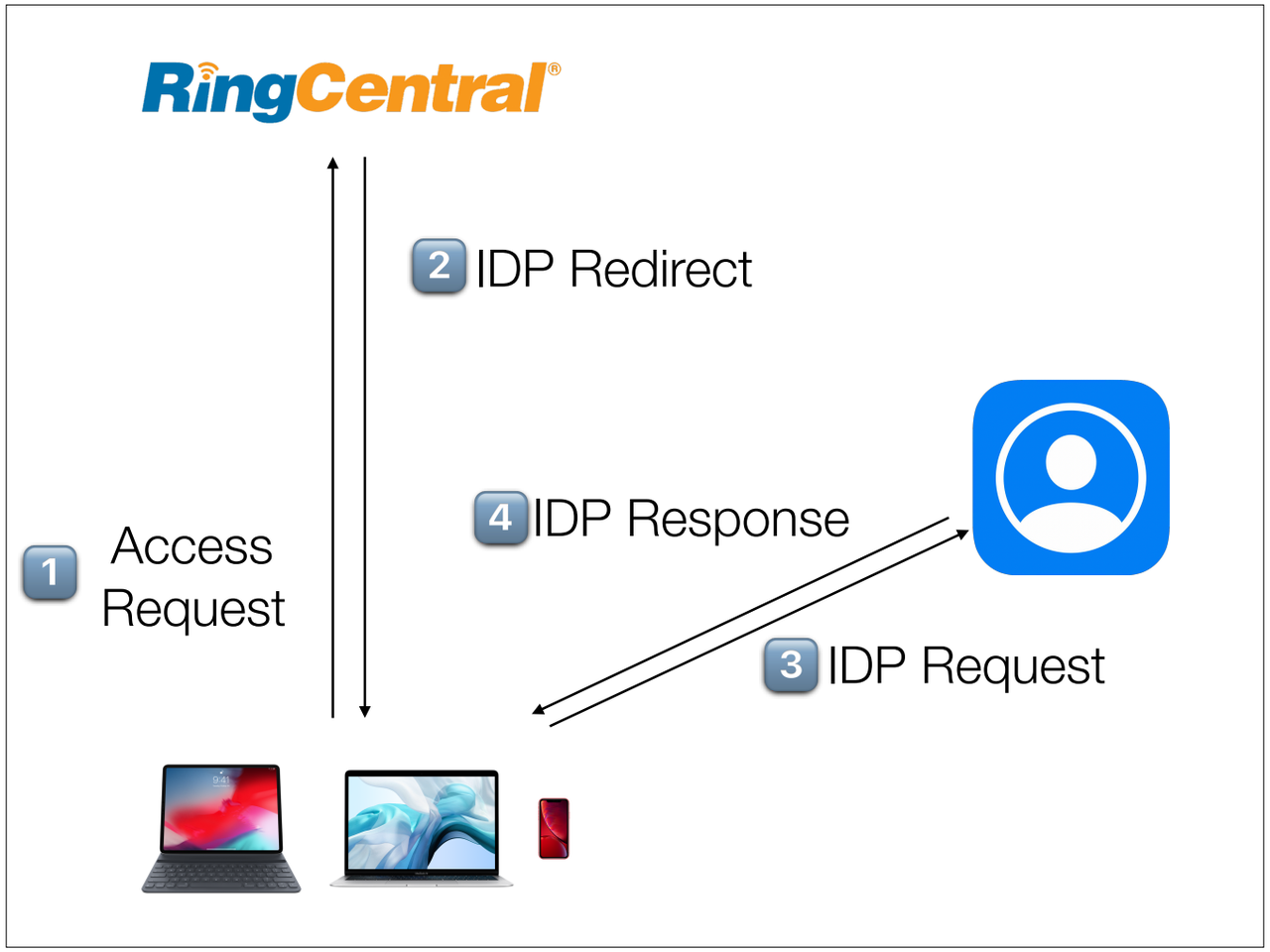


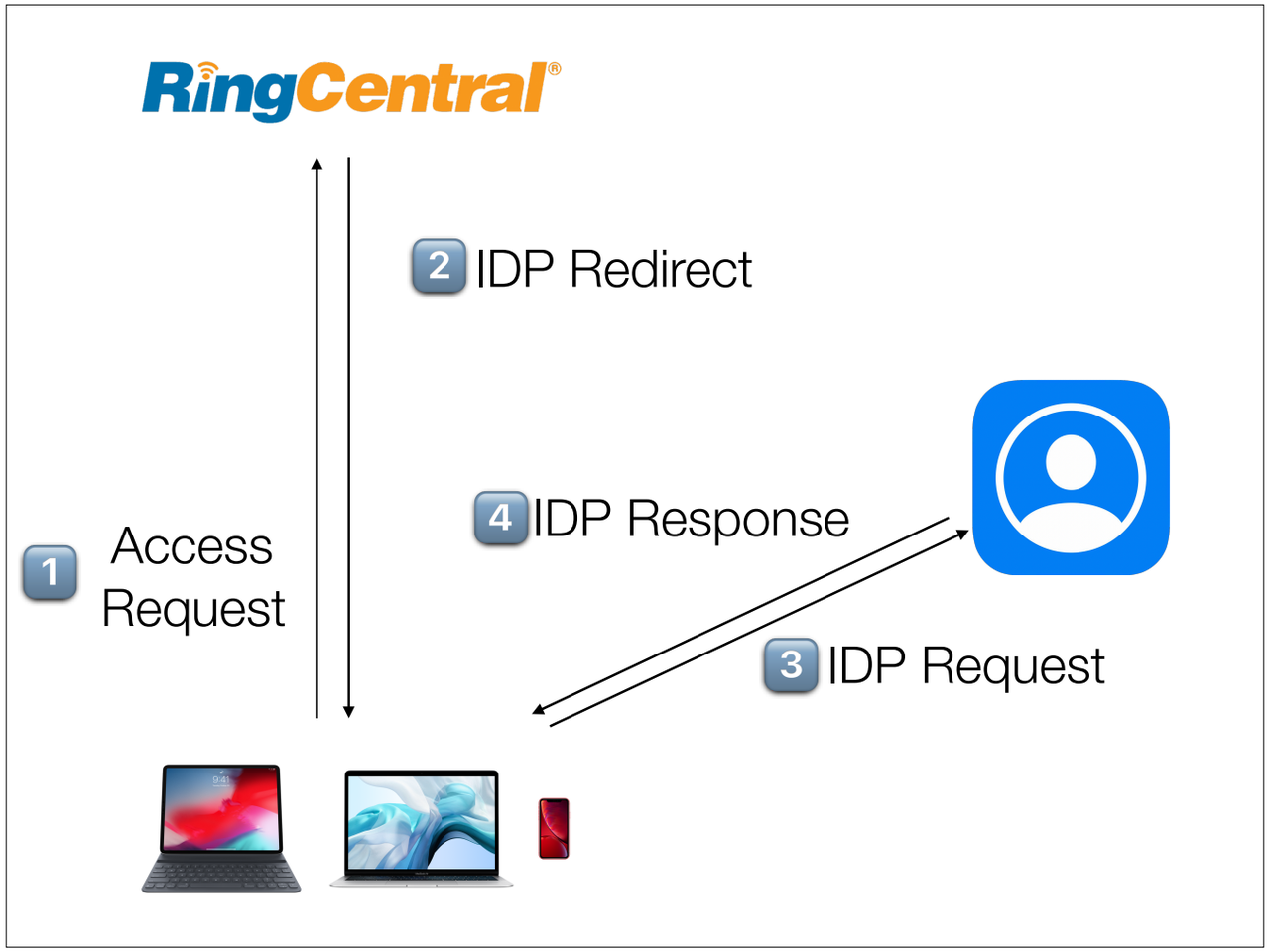
The user will so be redirected to the Identity Provider to get authenticated by any methods.


RingCentral
2 IDP Redirect
1 Access Request
3 IDP Request

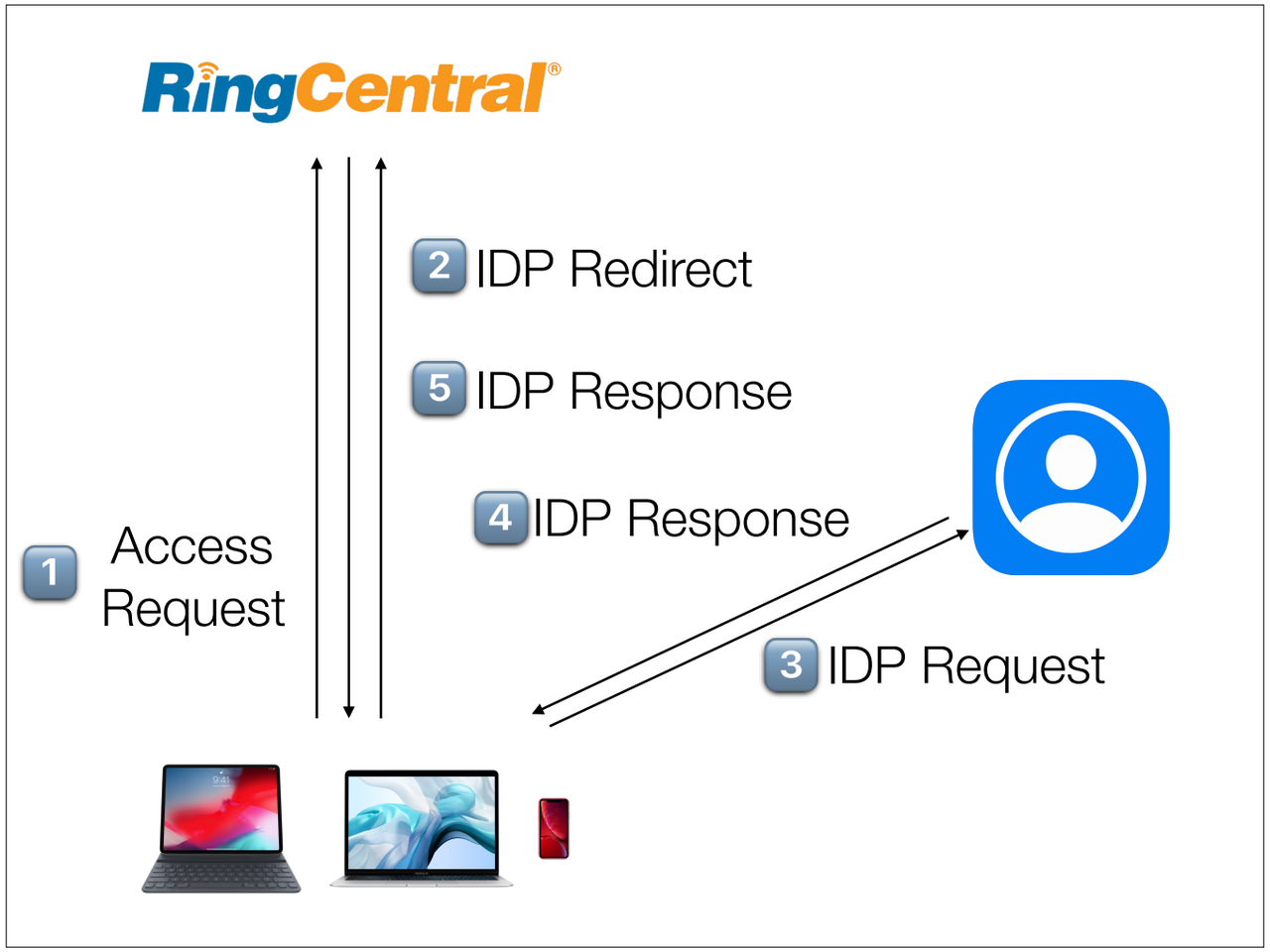


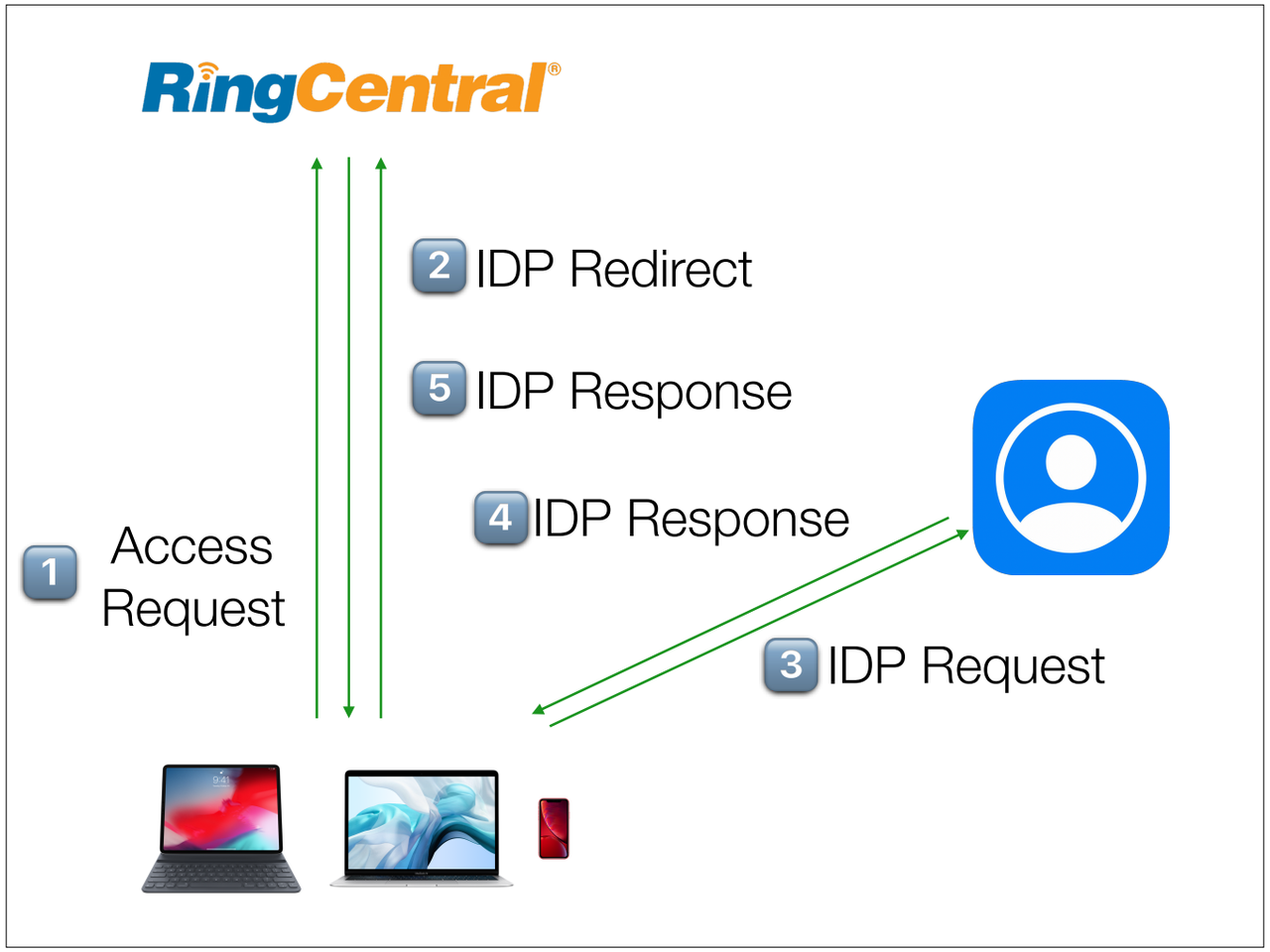
Once authenticated, the Identity Provider send an specific payload as response


RingCentral
2 IDP Redirect
1 Access Request
4 IDP Response
3 IDP Request

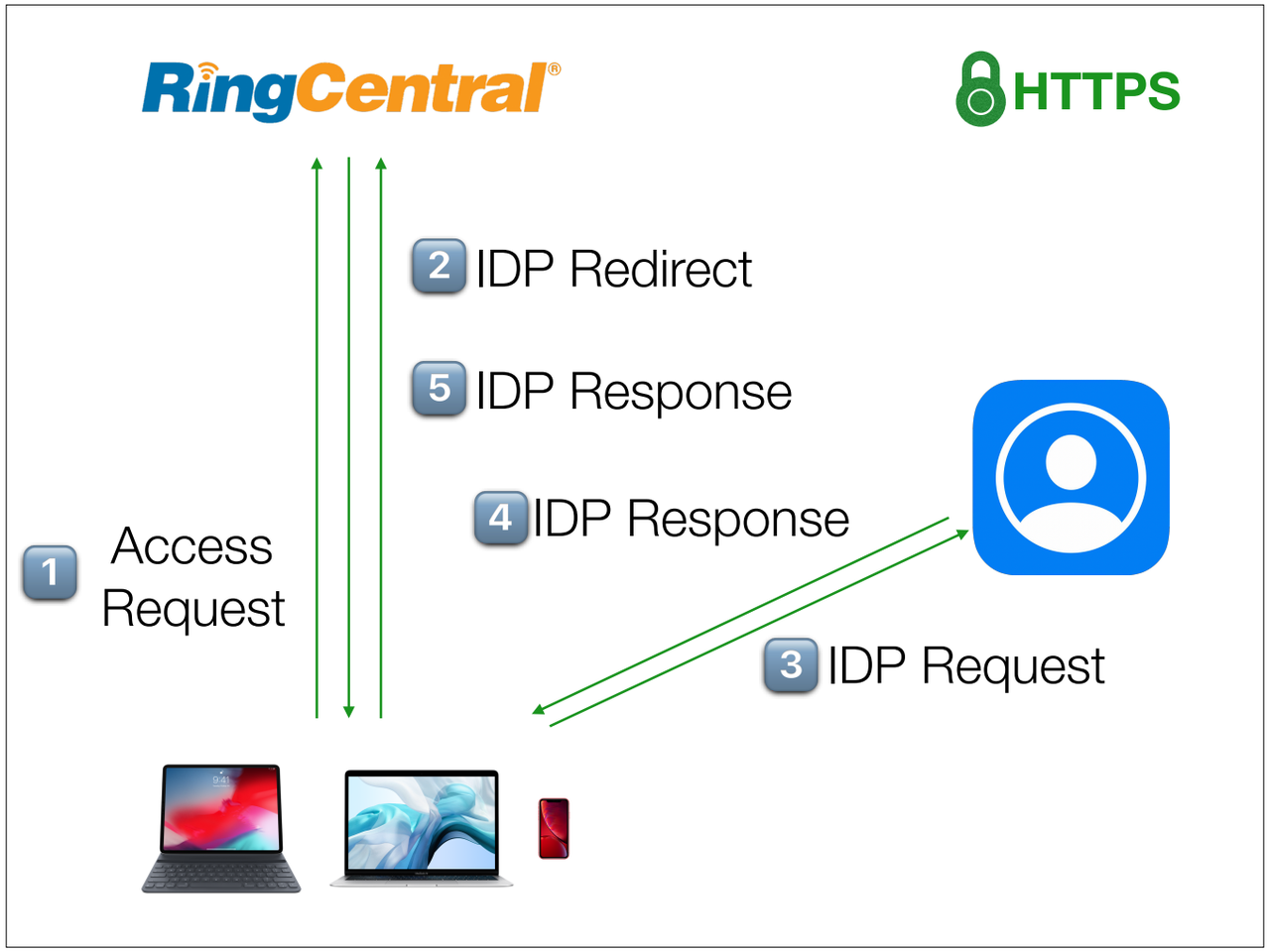


And this response will be forwarded back to the Service Provider to finally get access to the service the user want


RingCentral
2 IDP Redirect
5 IDP Response
4 IDP Response
1 Access Request
3 IDP Request



And of course, all of this is protected by HTTPS


RingCentral
HTTPS
2 IDP Redirect
5 IDP Response
4 IDP Response
1 Access Request
3 IDP Request

SAML Response

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" Destination="https://ssoeuro.ringcentral.com/sp/ACS.saml2" ID="_e5203c6345a72e6e5680c40fb6d7ce1f" IssueInstant="2019-06-30T20:14:29.972Z"
Version="2.0">
   <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">https://idp.storymaker.fr/md/idp.xml</saml:Issuer>
   <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
         <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
         <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
         <ds:Reference URI="#_e5203c6345a72e6e5680c40fb6d7ce1f">
            <ds:Transforms>
               <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
               <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
            <ds:DigestValue>...</ds:DigestValue>
         </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>...</ds:SignatureValue>
   </ds:Signature>
   <samlp:Status>
      <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success" />
   </samlp:Status>
   <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="_a3afb9a0c7da98beaa3fc19bedfc0db5" IssueInstant="2019-06-30T20:14:29.972Z" Version="2.0">
      <saml:Issuer>https://idp.storymaker.fr/md/idp.xml</saml:Issuer>
      <saml:Subject>
         <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress" NameQualifier="https://idp.storymaker.fr/md/idp.xml">alice@storymaker.fr</saml:NameID>
         <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
            <saml:SubjectConfirmationData NotOnOrAfter="2019-06-30T20:17:49.972Z" Recipient="https://ssoeuro.ringcentral.com/sp/ACS.saml2" />
         </saml:SubjectConfirmation>
      </saml:Subject>
      <saml:Conditions NotBefore="2019-06-30T20:14:14.972Z" NotOnOrAfter="2019-06-30T20:17:49.972Z">
         <saml:AudienceRestriction>
            <saml:Audience>saml2:ringcentral:prodeuro</saml:Audience>
         </saml:AudienceRestriction>
      </saml:Conditions>
      <saml:AuthnStatement AuthnInstant="2019-06-30T20:14:29.972Z" SessionIndex="_f81d1fe9b2fcc721af9badad89f386d7">
         <saml:AuthnContext>
            <saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</saml:AuthnContextClassRef>
         </saml:AuthnContext>
      </saml:AuthnStatement>
   </saml:Assertion>
</samlp:Response>
```

This IDP response, if we take SAML, look like that. Let’s start to read it

SAML Response

```
<samlp:Response
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  Destination="https://ssoeuro.ringcentral.com/sp/ACS.saml2"
  ID="_e5203c6345a72e6e5680c40fb6d7ce1f"
  IssueInstant="2019-06-30T20:14:29.972Z"
  Version="2.0">
…
</samlp:Response>
```

In the first part, you will have information about the target service for whom this response has been made

SAML Response

```
<samlp:Response
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  Destination="https://ssoeuro.ringcentral.com/sp/ACS.saml2"
  ID="_e5203c6345a72e6e5680c40fb6d7ce1f"
  IssueInstant="2019-06-30T20:14:29.972Z"
  Version="2.0">
…
</samlp:Response>
```

SAML Response

```
<saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">https://idp.storymaker.fr/md/idp.xml</saml:Issuer>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
   <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
      <ds:Reference URI="#_e5203c6345a72e6e5680c40fb6d7ce1f">
         <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
         </ds:Transforms>
         <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
         <ds:DigestValue>...</ds:DigestValue>
      </ds:Reference>
   </ds:SignedInfo>
   <ds:SignatureValue>...</ds:SignatureValue>
</ds:Signature>
<samlp:Status>
   <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success" />
</samlp:Status>
```

You will also have a signature part, to be sure that the response hasn’t been tempered in-between. It’s a proof of authentication so it has to be robust!

SAML Response

```
<saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">https://idp.storymaker.fr/md/idp.xml</saml:Issuer>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
   <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
      <ds:Reference URI="#_e5203c6345a72e6e5680c40fb6d7ce1f">
         <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
         </ds:Transforms>
         <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
         <ds:DigestValue>...</ds:DigestValue>
      </ds:Reference>
   </ds:SignedInfo>
   <ds:SignatureValue>...</ds:SignatureValue>
</ds:Signature>
<samlp:Status>
   <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success" />
</samlp:Status>
```

SAML Response

```
<saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress"
  NameQualifier="https://idp.storymaker.fr/md/idp.xml">
    alice@storymaker.fr
</saml:NameID>
```

And then you get the NameID, the main identifier related to the user identity, it can be an e-mail for example.

SAML Response

```
<saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress"
  NameQualifier="https://idp.storymaker.fr/md/idp.xml">
   alice@storymaker.fr
</saml:NameID>
```

SAML Response

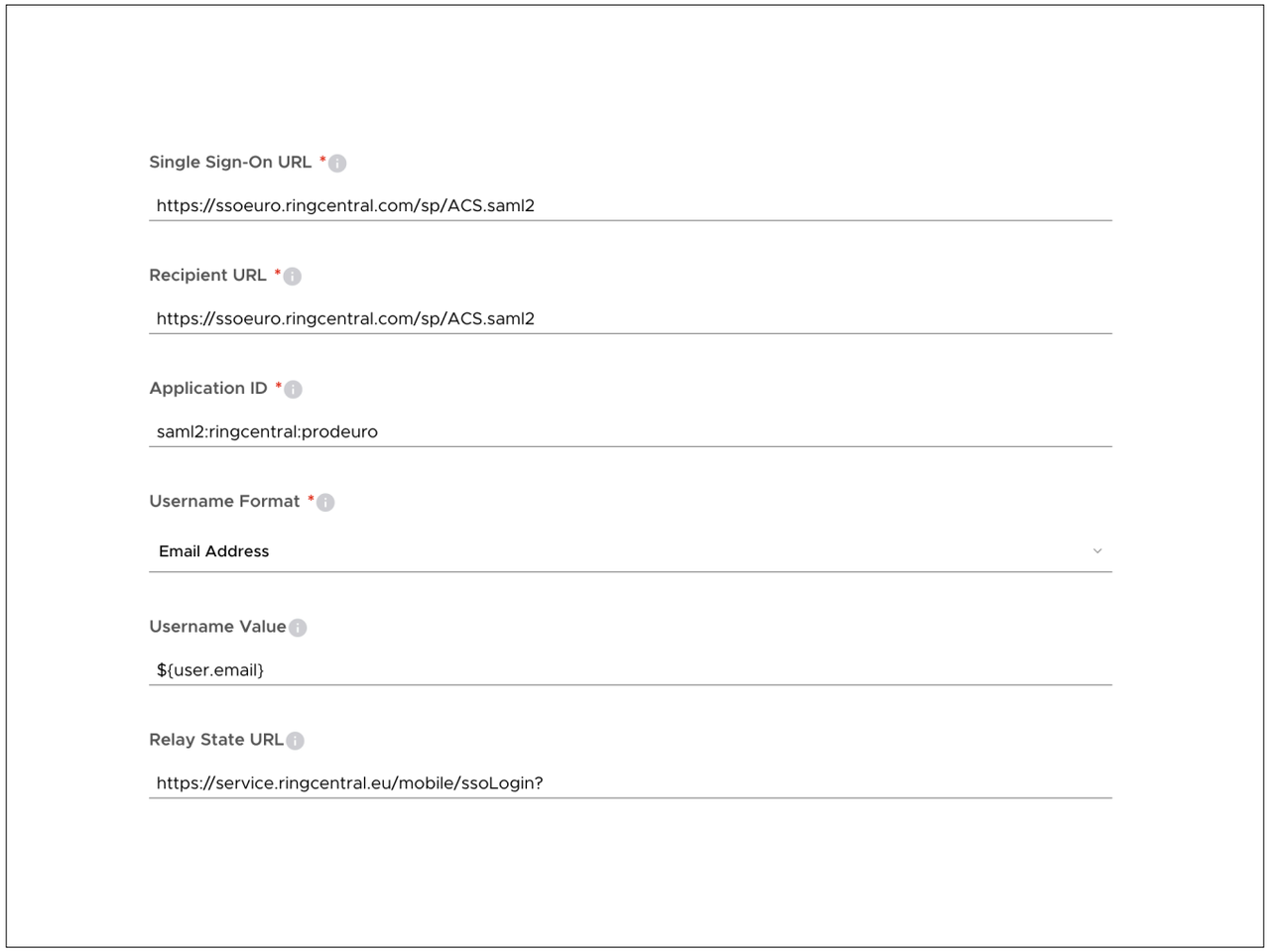
```
<saml:AttributeStatement>
        <saml:Attribute AttributeName="EMail">
                <saml:AttributeValue xsi:type="xsd:string">alice@storymaker.fr</saml:AttributeValue>
        </saml:Attribute>
        <saml:Attribute AttributeName="ImmutableID">
                <saml:AttributeValue xsi:type="xsd:string">EEFF7781-A1A4-47C9-A5C5-767AE358AB9A</saml:AttributeValue>
        </saml:Attribute>
        <saml:Attribute AttributeName="FirstName">
                <saml:AttributeValue xsi:type="xsd:string">Alice</saml:AttributeValue>
        </saml:Attribute>
        <saml:Attribute AttributeName="LastName">
                <saml:AttributeValue xsi:type="xsd:string">Smith</saml:AttributeValue>
        </saml:Attribute>
</saml:AttributeStatement>
```

And you can also have additional attribute to provide additional info like:

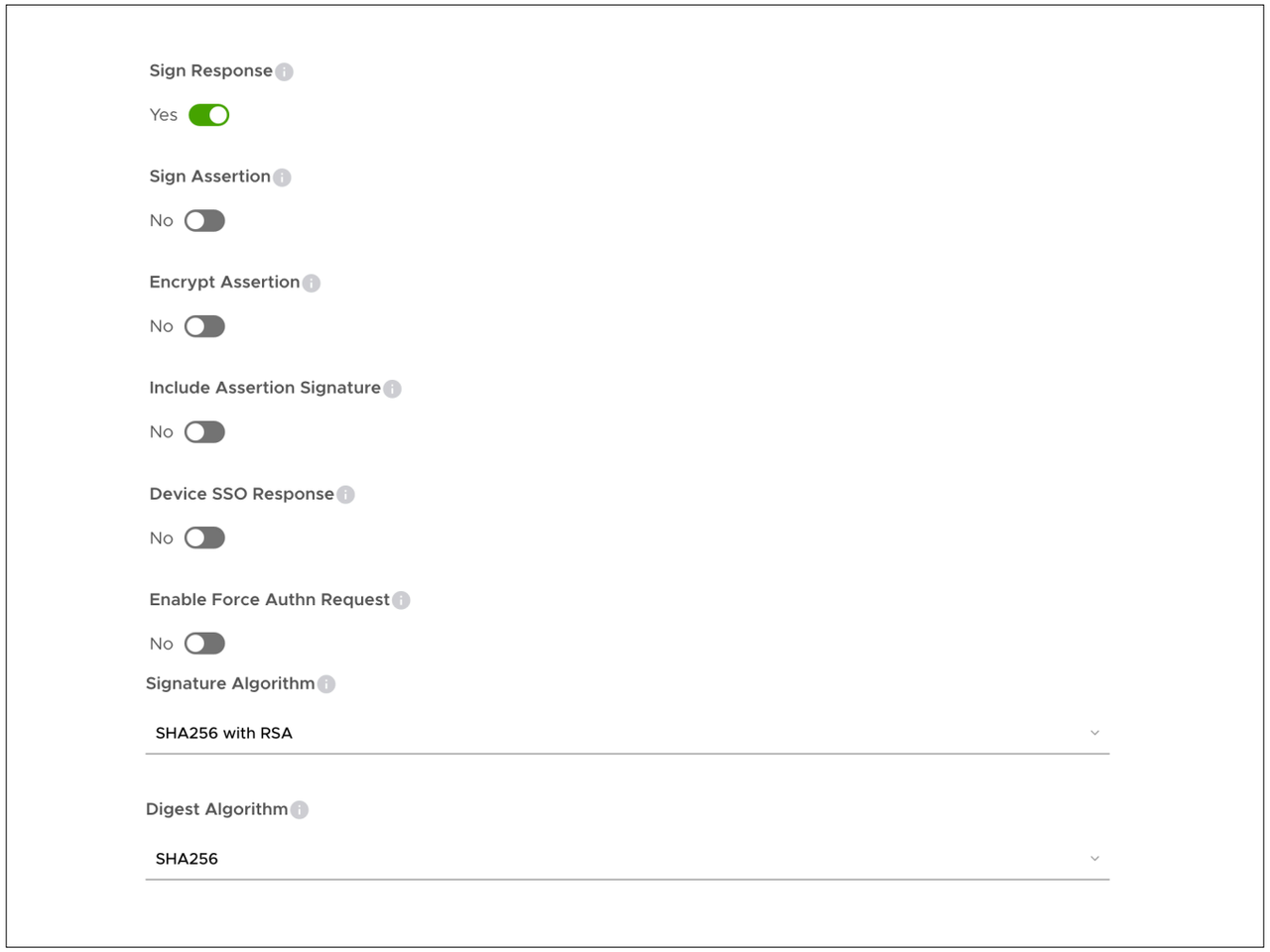
- the e-mail address
- eventually a source anchor helping the Service Provider to keep track of the user record when people change their name
- the current first name
- the current last name

SAML Response

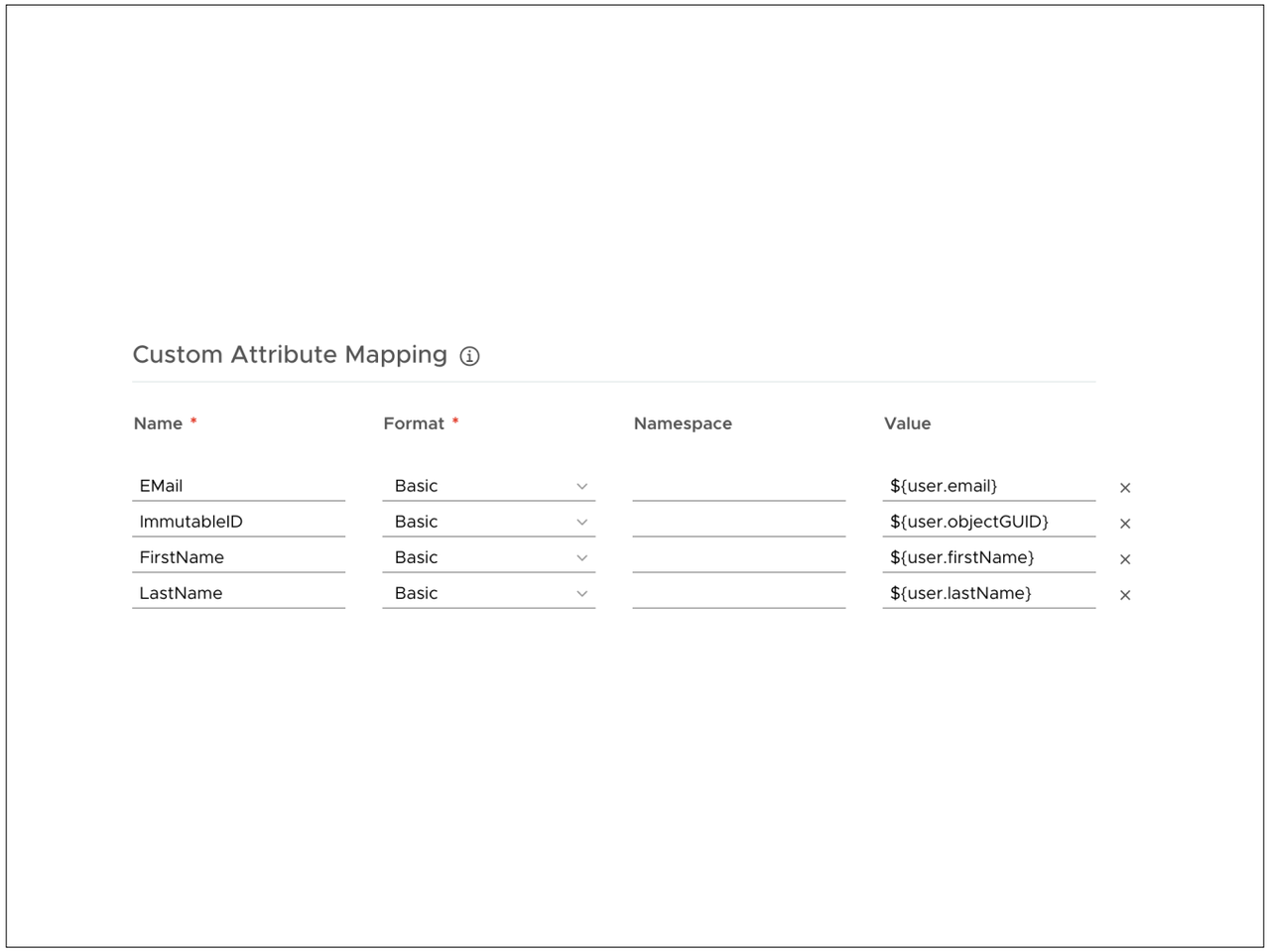
```
<saml:AttributeStatement>
        <saml:Attribute AttributeName="EMail">
                <saml:AttributeValue xsi:type="xsd:string">alice@storymaker.fr</saml:AttributeValue>
        </saml:Attribute>
        <saml:Attribute AttributeName="ImmutableID">
                <saml:AttributeValue xsi:type="xsd:string">EEFF7781-A1A4-47C9-A5C5-767AE358AB9A</saml:AttributeValue>
        </saml:Attribute>
        <saml:Attribute AttributeName="FirstName">
                <saml:AttributeValue xsi:type="xsd:string">Alice</saml:AttributeValue>
        </saml:Attribute>
        <saml:Attribute AttributeName="LastName">
                <saml:AttributeValue xsi:type="xsd:string">Smith</saml:AttributeValue>
        </saml:Attribute>
</saml:AttributeStatement>
```



Configuring such a link on the Identity Provider side, you will have to fill it with values provided by the Service Provider such as the target link or the kind of NameID needed



Then you will have to set the signature and encryptions pattern as requested by the Service Provider



And finally, the additional attributes you want to provide, here at the left the attribute name in the SAML answer, and on the right the attribute to use from your user database

Modern User Provisioning

SCIM

For User Provisioning, there is way less solution: it’s all about SCIM

SCIM

Modern directory query language

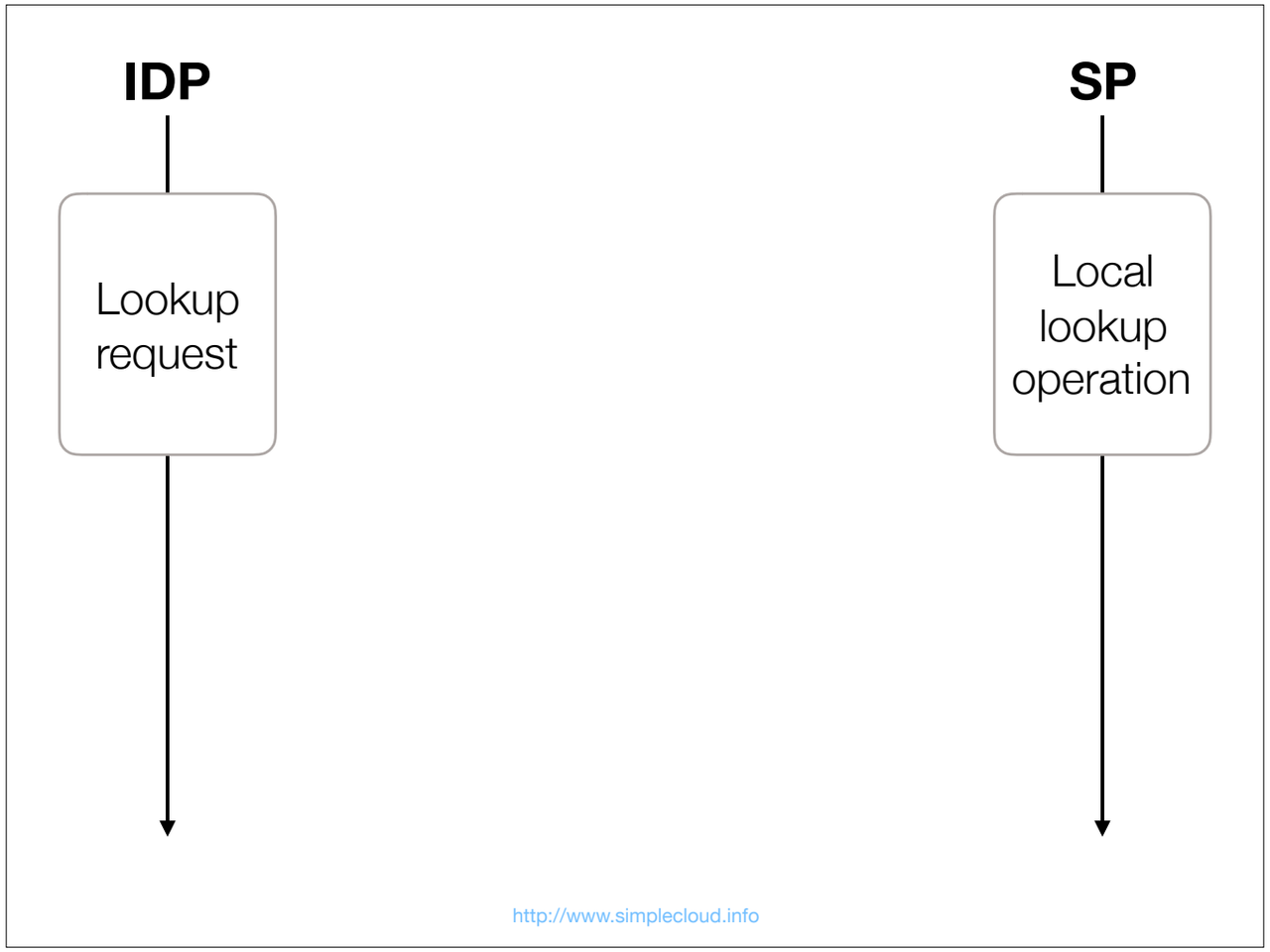
"modern" aka JSON over HTTPS

Inverted operation: IDP to SP

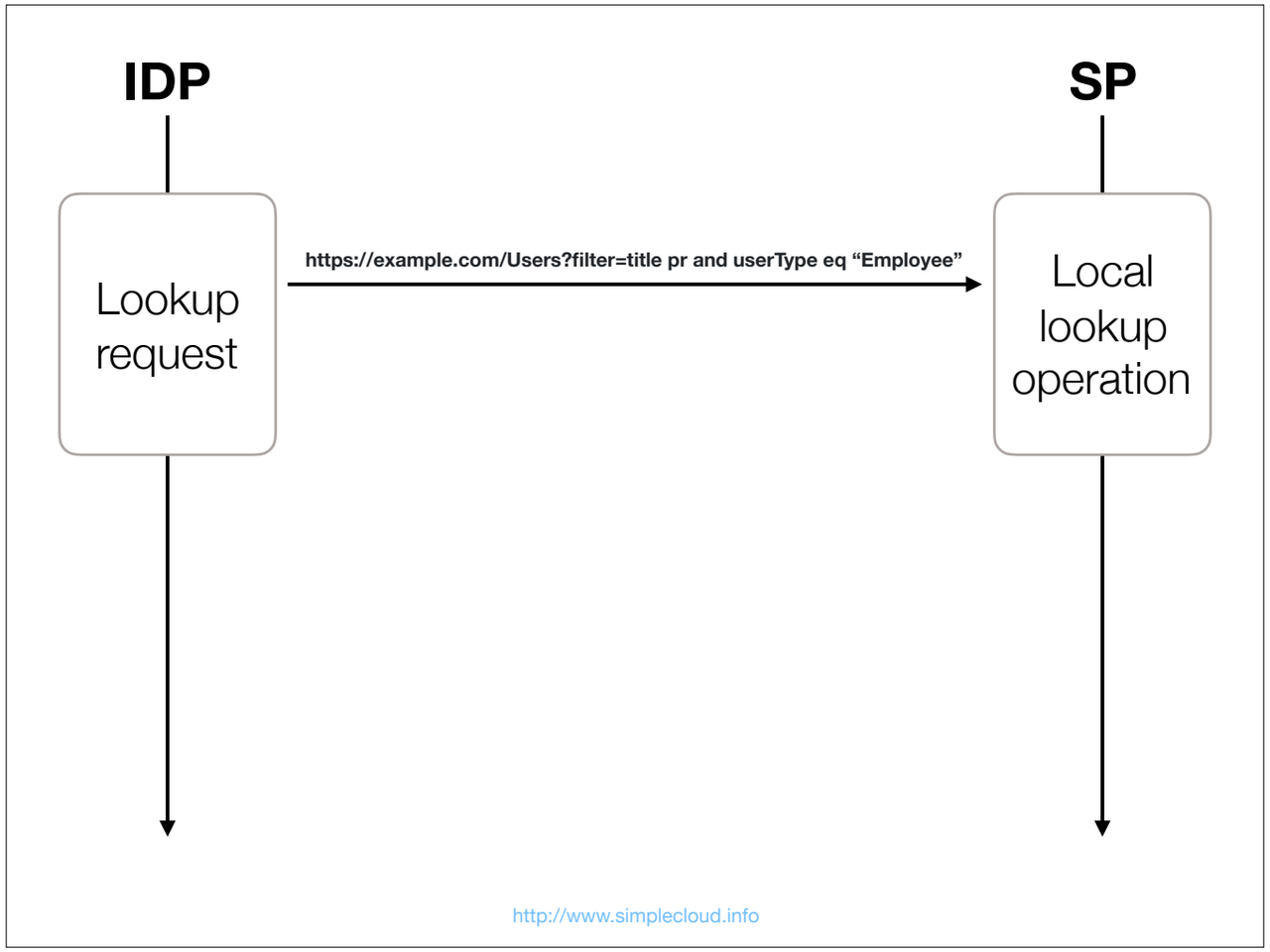
Allow usual CURD operation

Support varieties of search filters

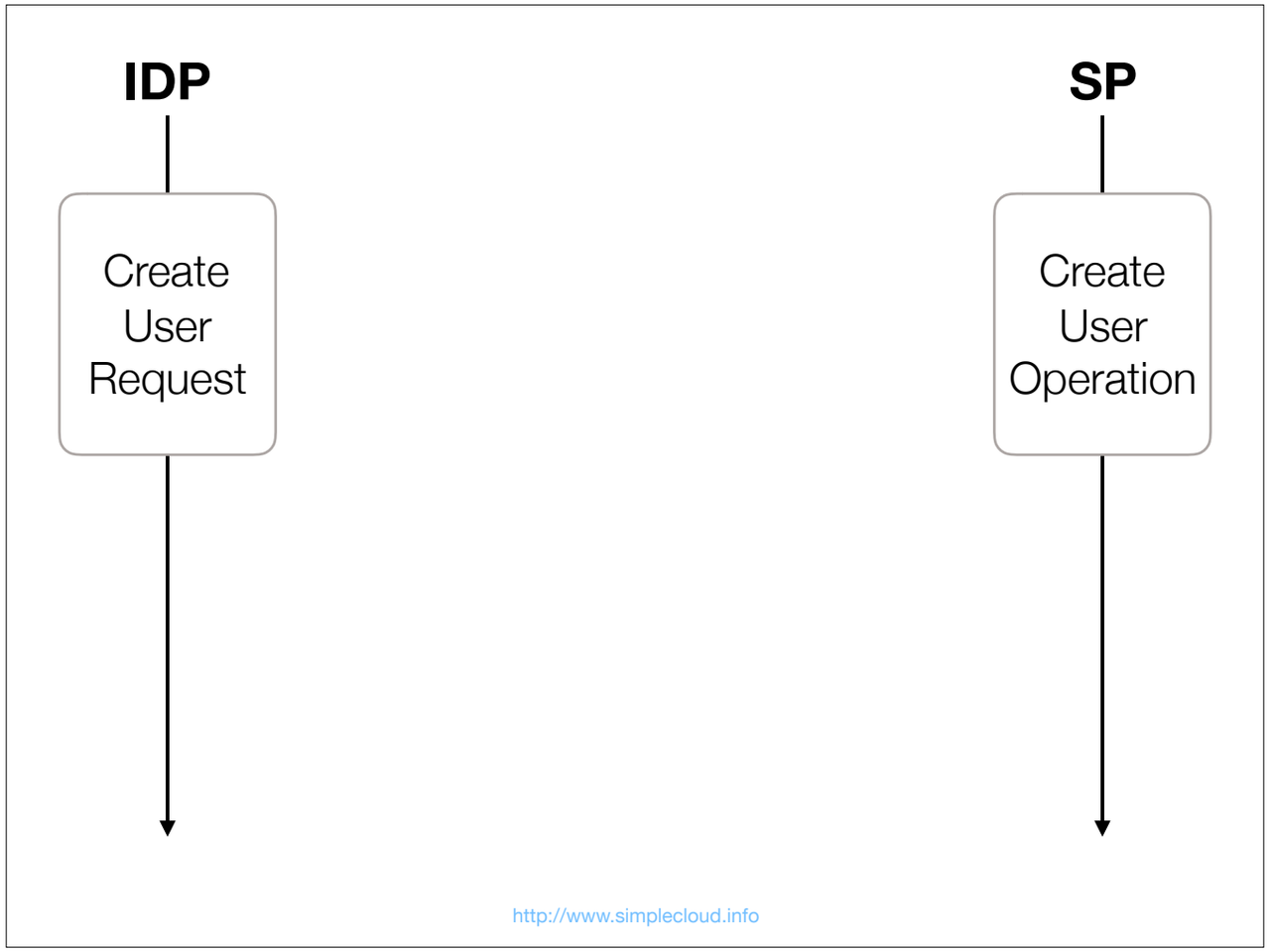
SCIM is a modern directory querying language (aka JSON over HTTP) but with reversed operation, the Identity Provider will use it to request the Service Provider for its current state, and then support all regular Create Update Read and Delete operation to push change from the IDP to the SP.



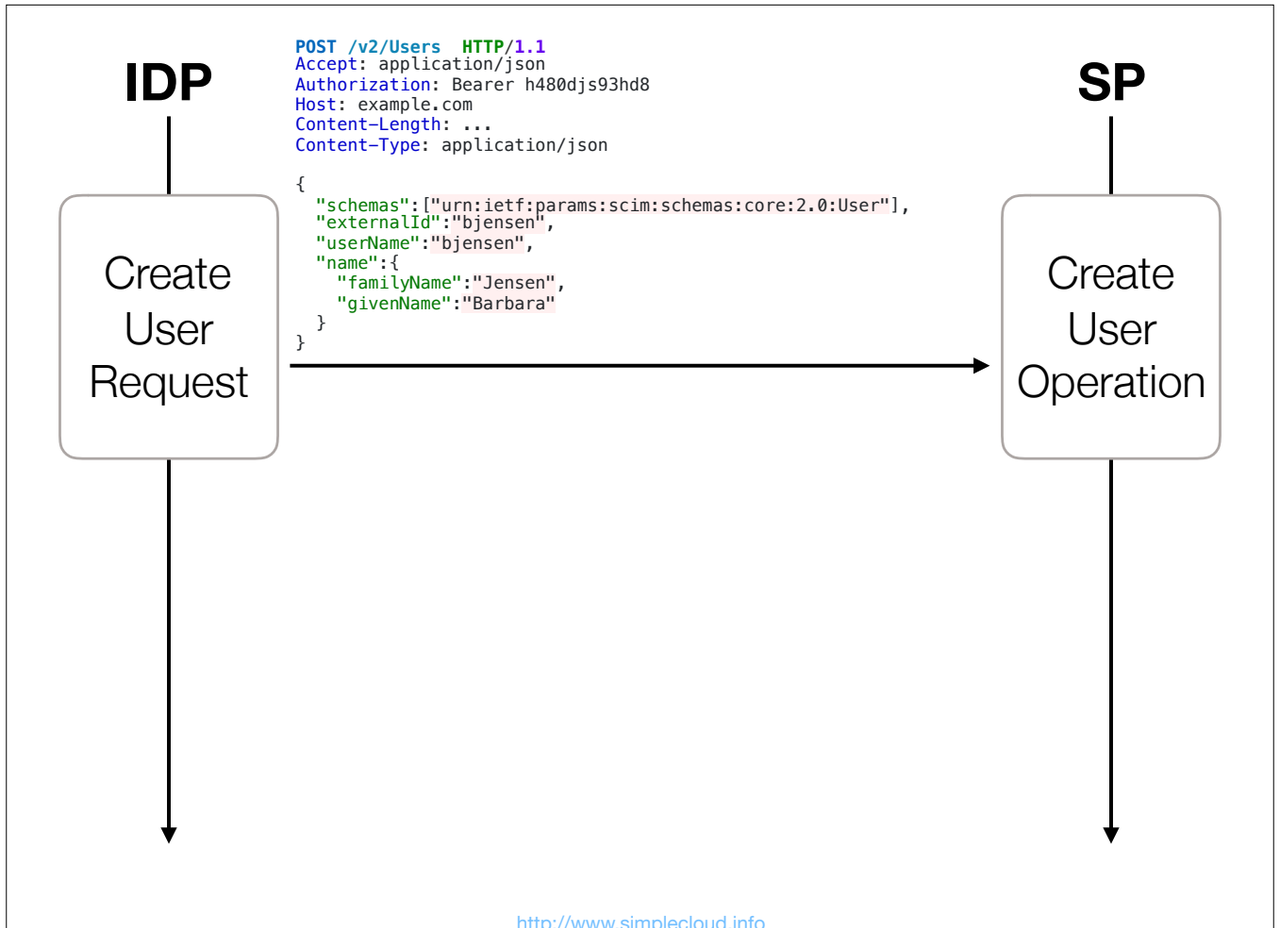
Here is an example, the IDP doing a lookup request to the SP via a simple REST call containing a filter string, and the SP to answer with a JSON formatted related record

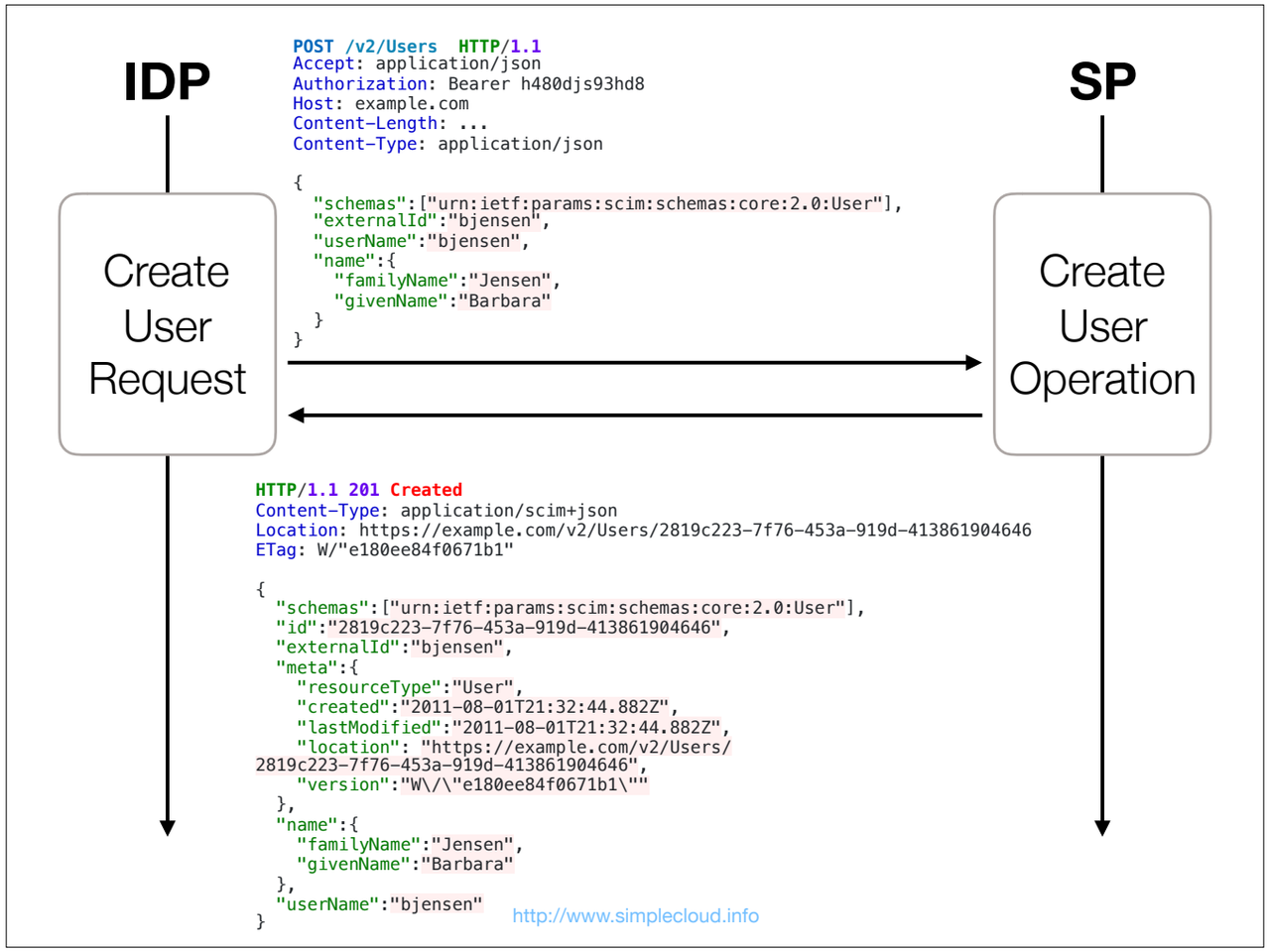

IDP
SP
Lookup request
https://example.com/Users?filter=title pr and userType eq “Employee”
Local lookup operation
http://www.simplecloud.info

http://www.simplecloud.info

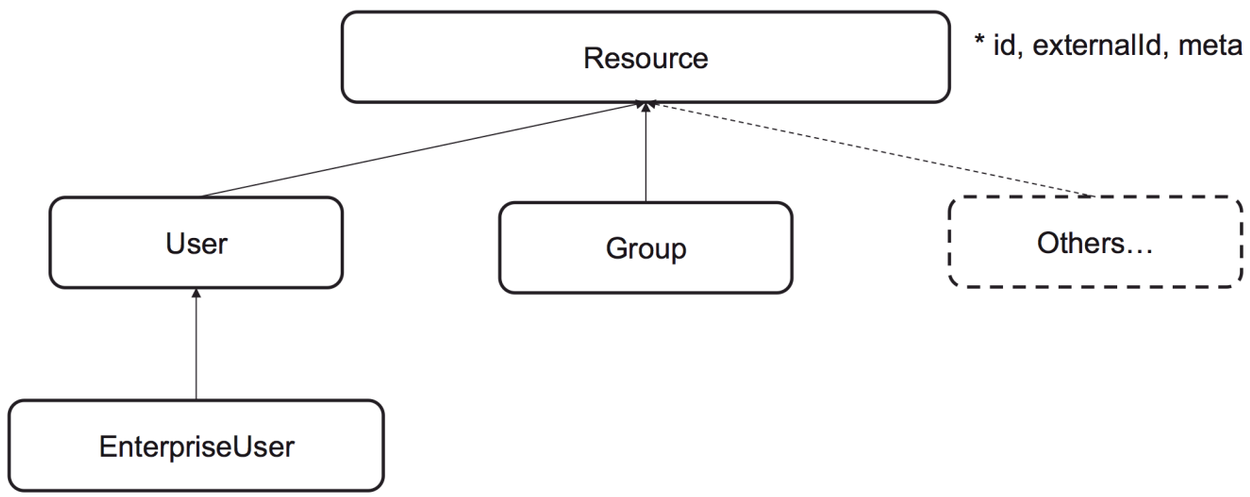


When the IDP want to push a change, it will simply do a POST request to the REST API with the JSON payload, and the SP to answer with the newly created object.


IDP
SP
POST /v2/Users HTTP/1.1
Accept: application/json
Authorization: Bearer h480djs93hd8
Host: example.com
Content-Length: ...
Content-Type: application/json
{
"schemas":["urn:ietf:params:scim:schemas:core:2.0:User"],
"externalId":"bjensen",
"userName":"bjensen",
"name":{
"familyName":"Jensen",
"givenName":"Barbara"
}
}
Create User Request
Create User Operation
http://www.simplecloud.info


IDP
SP
POST /v2/Users HTTP/1.1
Accept: application/json
Authorization: Bearer h480djs93hd8
Host: example.com
Content-Length: ...
Content-Type: application/json
{
"schemas":["urn:ietf:params:scim:schemas:core:2.0:User"],
"externalId":"bjensen",
"userName":"bjensen",
"name":{
"familyName":"Jensen",
"givenName":"Barbara"
}
}
Create User Request
Create User Operation
HTTP/1.1 201 Created
Content-Type: application/scim+json
Location: https://example.com/v2/Users/2819c223-7f76-453a-919d-413861904646
ETag: W/"e180ee84f0671b1"
{
"schemas":["urn:ietf:params:scim:schemas:core:2.0:User"],
"id":"2819c223-7f76-453a-919d-413861904646",
"externalId":"bjensen",
"meta":{
"resourceType":"User",
"created":"2011-08-01T21:32:44.882Z",
"lastModified":"2011-08-01T21:32:44.882Z",
"location": "https://example.com/v2/Users/
2819c223-7f76-453a-919d-413861904646",
"version":"W\/\"e180ee84f0671b1\""
},
"name":{
"familyName":"Jensen",
"givenName":"Barbara"
},
"userName":"bjensen"
}
http://www.simplecloud.info

SCIM



http://www.simplecloud.info

SCIM is made to support user and group by default and can be extended as needed

SCIM

http://www.simplecloud.info

SCIM as a lot of built-in operations, all the one needed for directory operation and more

SCIM

Operation	Method	Sample URL
Create	POST	`https://example.com/{v}/{resource}`
Read	GET	`https://example.com/{v}/{resource}/{id}`
Replace	PUT	`https://example.com/{v}/{resource}/{id}`
Delete	DELETE	`https://example.com/{v}/{resource}/{id}`
Update	PATCH	`https://example.com/{v}/{resource}/{id}`
Search	GET	`https://example.com/{v}/{resource}?filter={attribute}{op}{value}&sortBy={attributeName}&sortOrder={ascending|descending}`
Bulk	POST	`https://example.com/{v}/Bulk`

http://www.simplecloud.info

SCIM, discovery

http://www.simplecloud.info

And also have discovery API so IDP can somehow be configured automatically once your give it the base URL of the SCIM SP API

SCIM, discovery

Path	Result
GET /ServiceProviderConfig	Specification compliance, authentication schemes, data models
GET /ResourceTypes	An endpoint used to discover the types of resources available
GET /Schemas	Introspect resources and attribute extensions

SCIM

Provisioning tools must abstract the complexity

Not a lot of compatible IDP and SP yet

But basically, you don’t really have to handle those things, it’s the IDP job to abstract this complexity.
The only issue is that too few IDP and SP are actually compatible with SCIM yet, so we have to push for it.

User Experience

SP or IDP initiated?

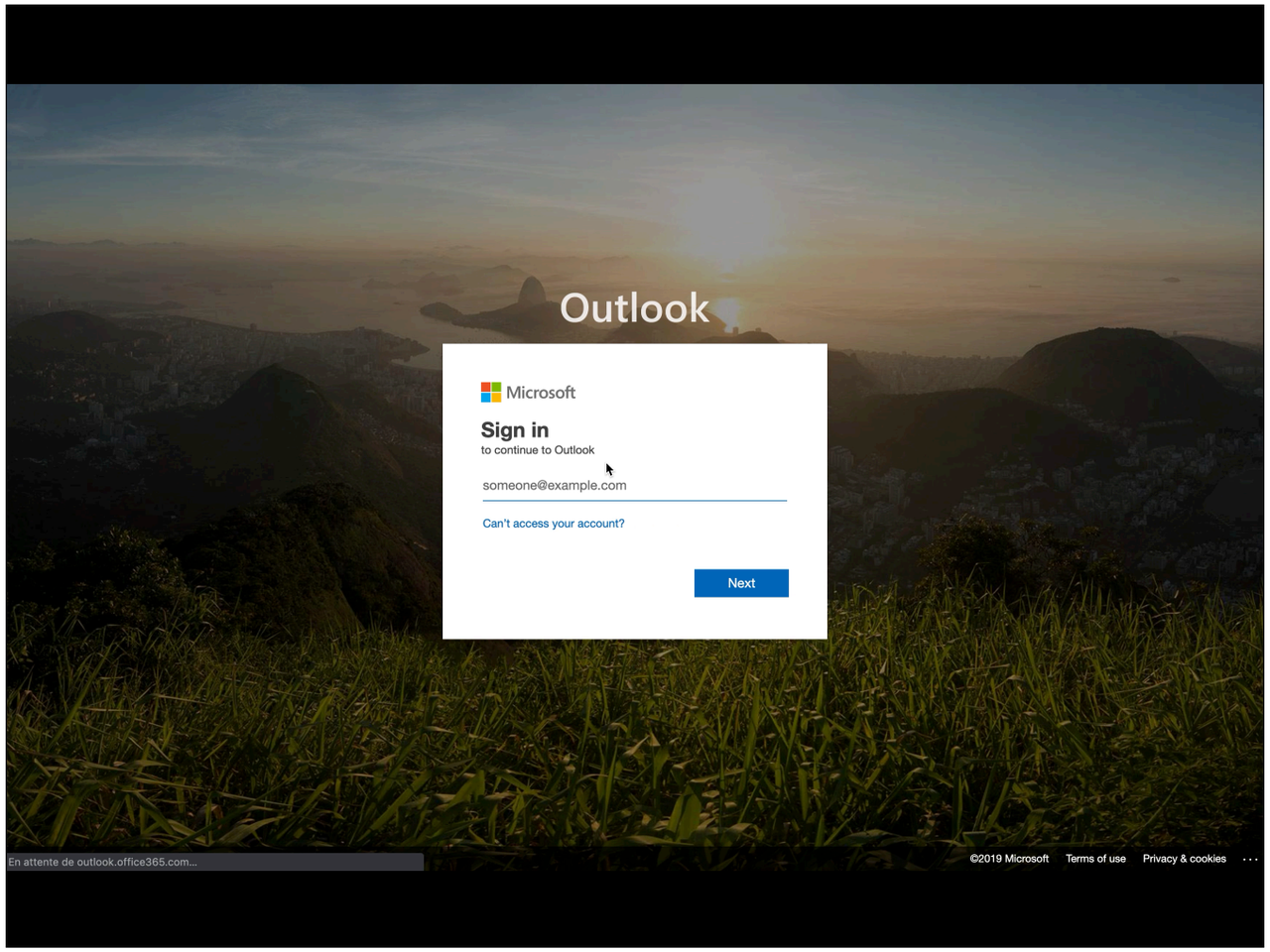
Let’s now consider you have your SP and IDP all set and move our focus to the User Experience

SP Initiated Logon

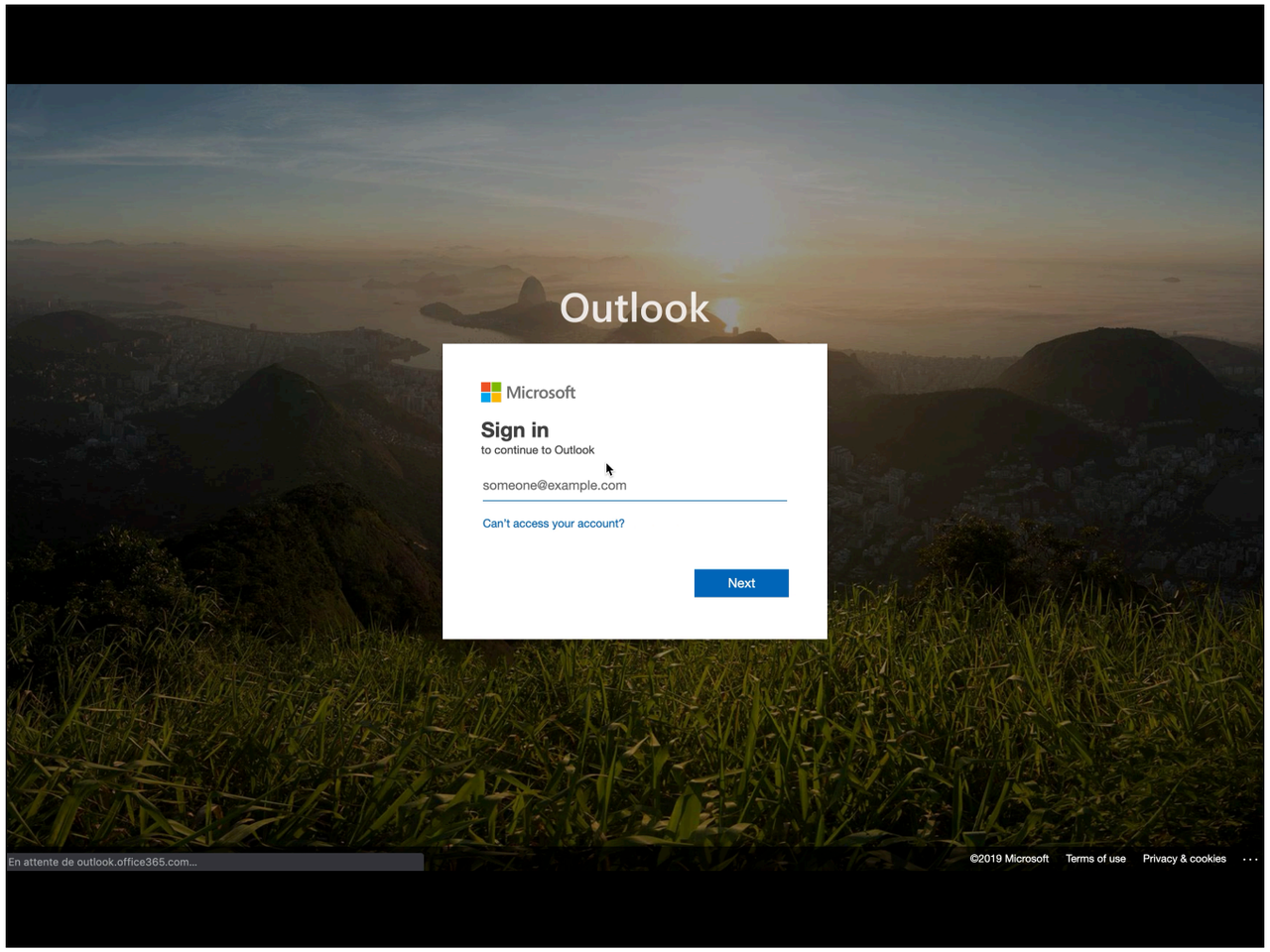
Users:

- Access the service provider
- Enter they e-mail address
- Get redirected to the identity provider
- Are requested to log in
- Are redirected with auth token to the service

The first User Experience I want to demonstrate is the « expected » one, when the login is SP Initiated: the user request the service and then get asked to go to the IDP for authentication purpose



Here is how it look

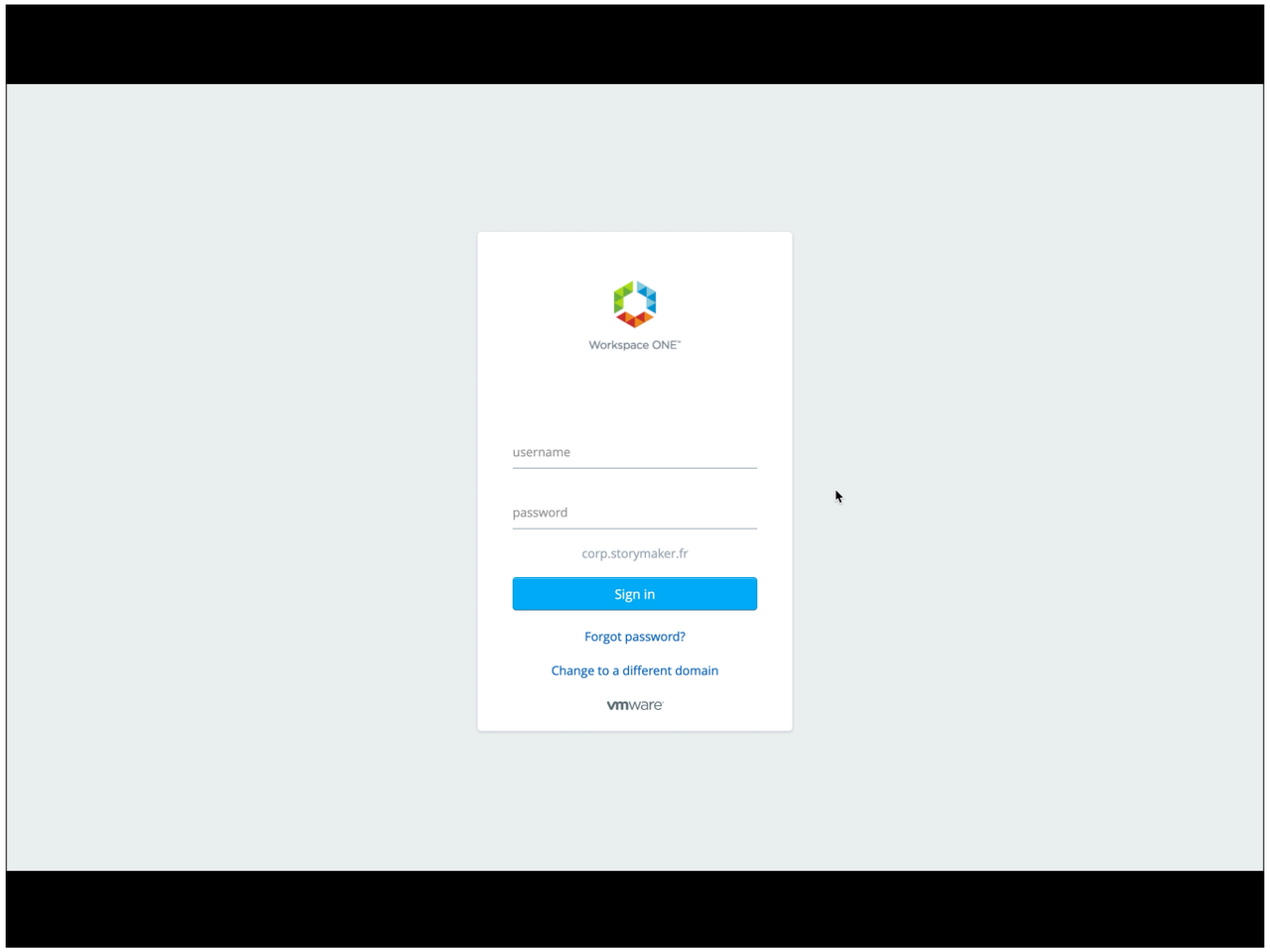

Outlook
Microsoft
Sign in
to continue to Outlook
someone@example.com
Can't access your account?
Next
En attente de outlook.office365.com...
©2019 Microsoft
Terms of use
Privacy & cookies
...

IDP Initiated Logon

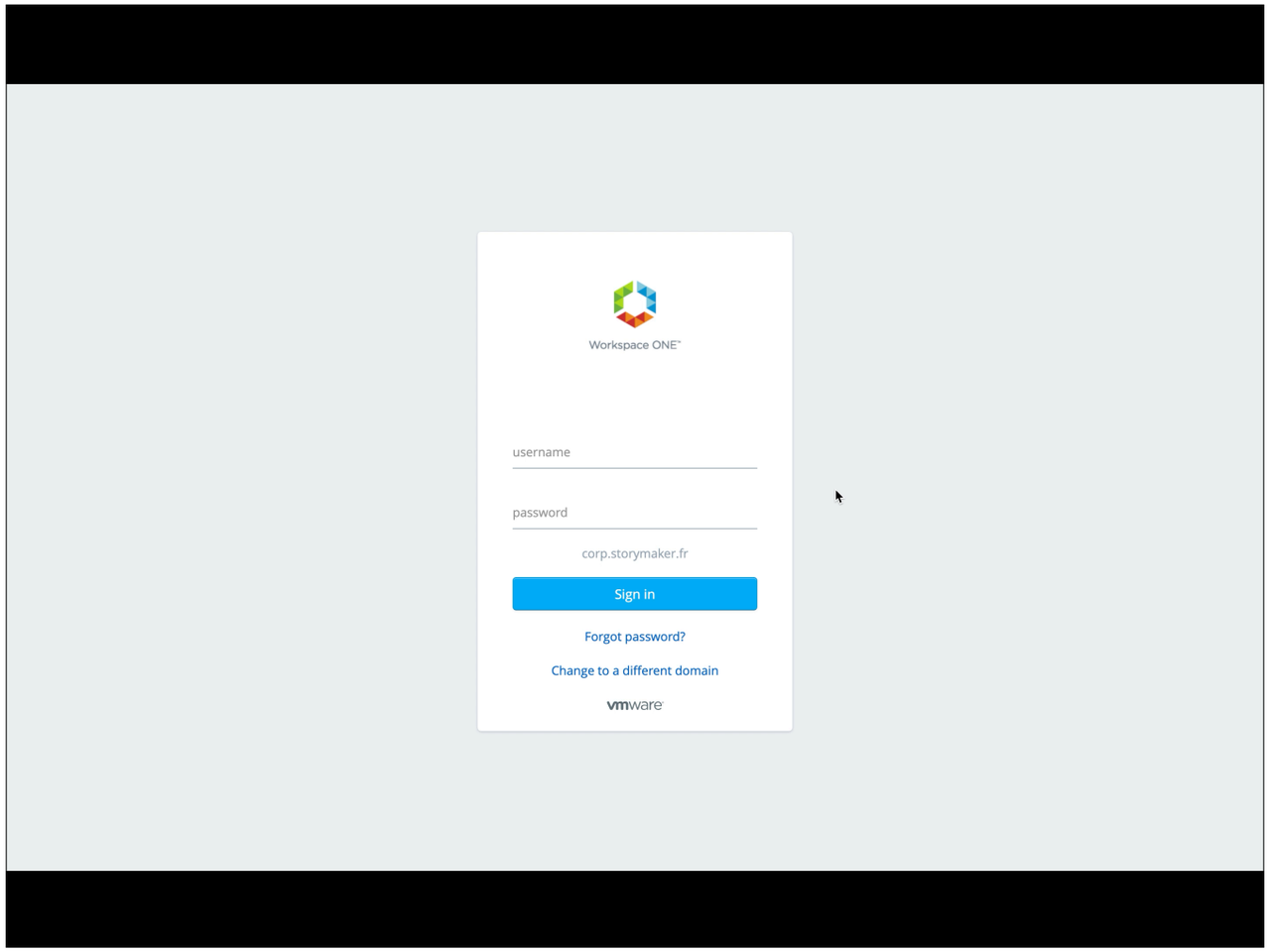
Users:

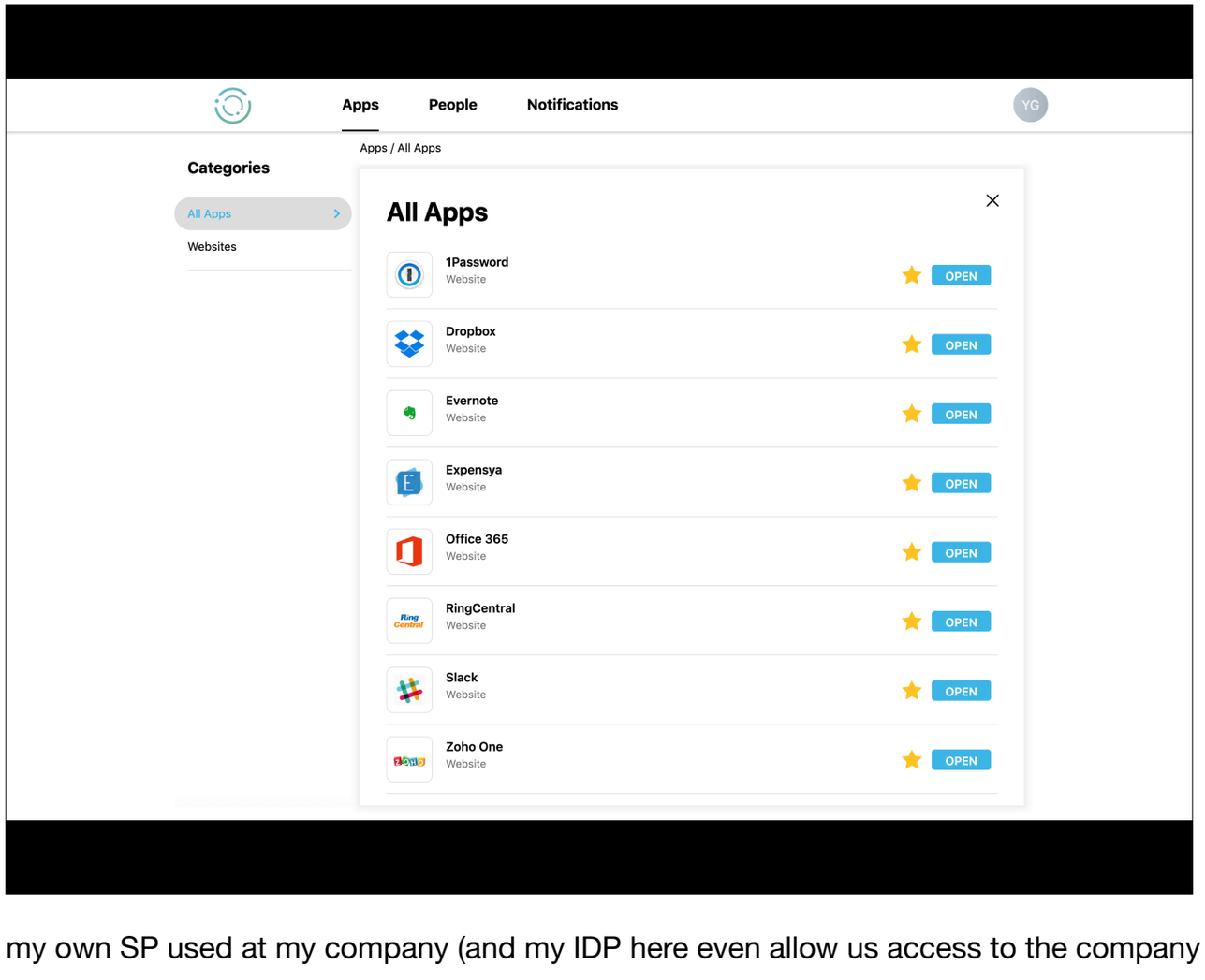
- Access the identity provider
- Are requested to log in
- Select the desired service
- Are redirected with auth token to the service

Now let’s talk about another way of doing this: the IDP initiated logon. The user go to the IDP, get authenticated, get a list of available service, and get connected to it



Here is how it look. And I think this is awesome. It help to reduce a lot the need for training and support for all services: train users once to get on your IDP launch page, and all upcoming services will be found there.


Workspace ONE™
username
password
corp.storymaker.fr
Sign in
Forgot password?
Change to a different domain
vmware



Here is the example of my own IDP with my own SP used at my company (and my IDP here even allow us access to the company directory to find contact method on our co-worker)

But there is more, we can provide a way better Corporate User Experience, and this « Corporate User Experience » instead of simple « User Experience » is something more and more important for me, it’s all about making a seamless experience when they use our solution for their works. And for that we will use Single Sign On.

What we’ve seen so far is Identity Federation: unique ID used for everything, but login and password still needed

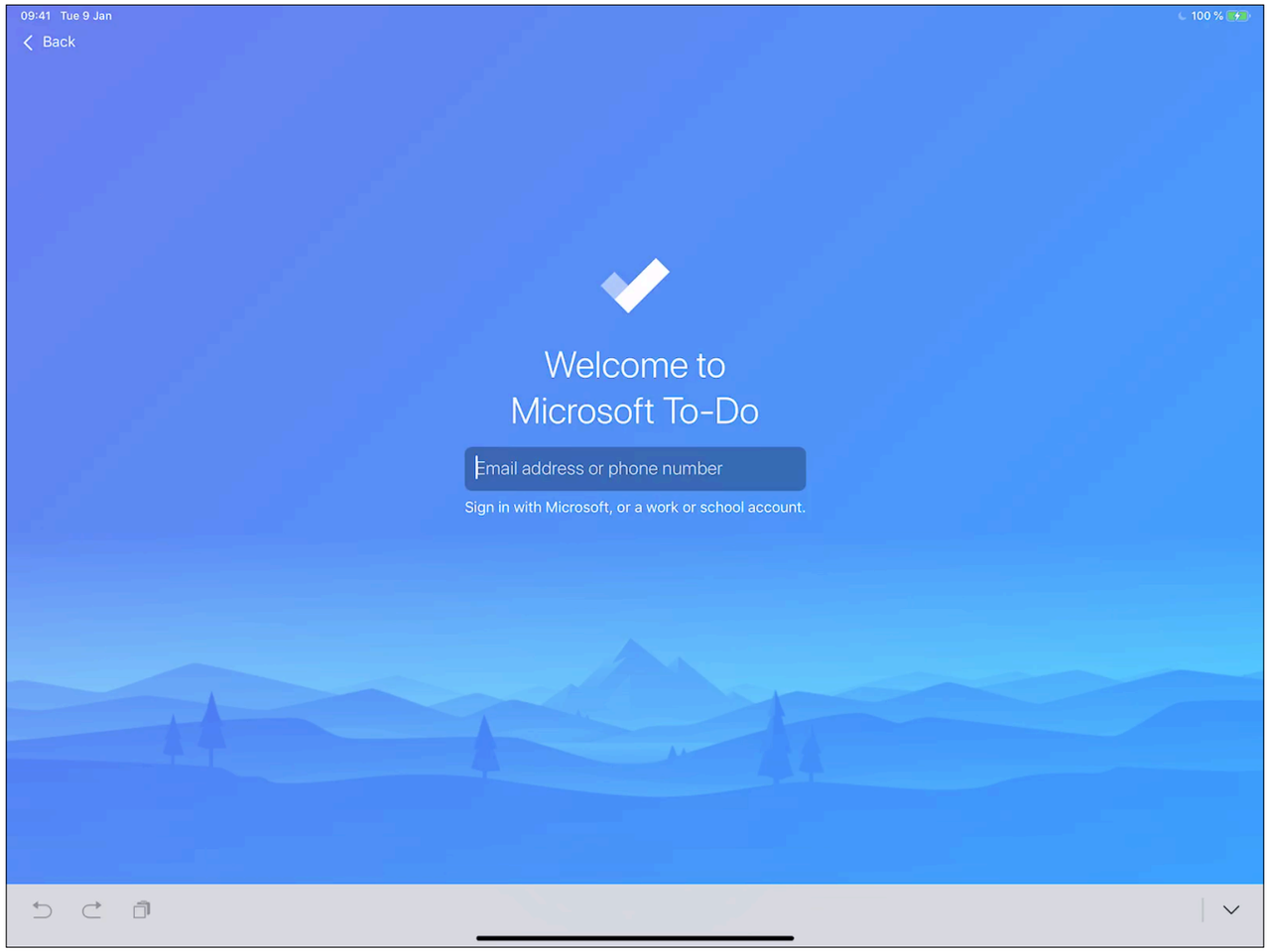
SSO on iOS

Working since iOS 7

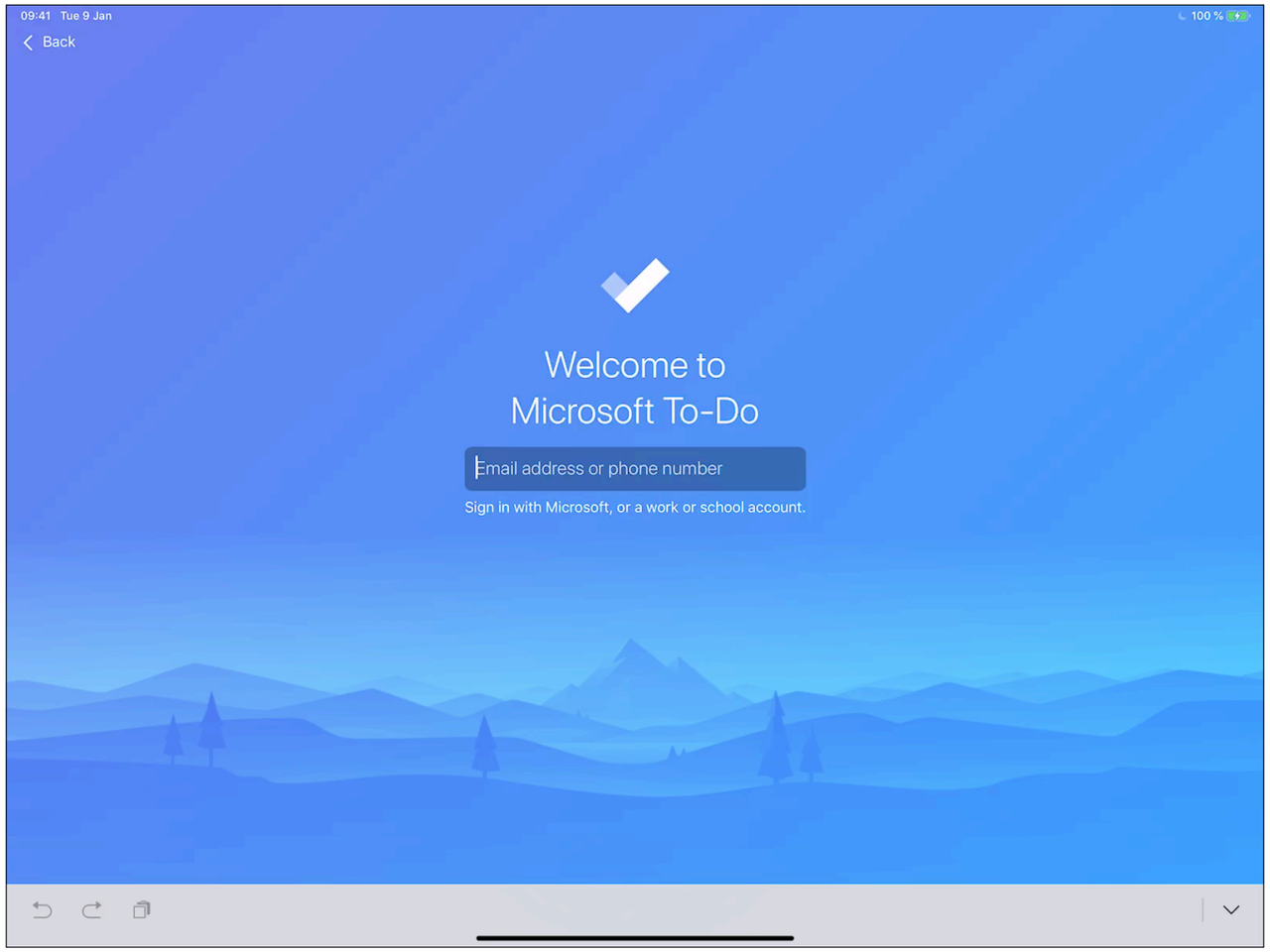
Based on public Kerberos realm

Fully provided by iOS, no developer work needed

SSO for iOS is available since iOS 7 and it works with a Kerberos realm that must be available in the public space (i.e. Internet). And all the work is done by iOS itself, Developers have nothing to handle.



Here is how it look on an iPad


09:41 Tue 9 Jan
100 %
Back
Welcome to
Microsoft To-Do
Email address or phone number
Sign in with Microsoft, or a work or school account.

iOS 13 SSO extension?

Not all IDP are good enough to handle Kerberos

Should extend the capabilities if your IDP is limited

Will require additional app

UX not as good if extension get installed last

So why there is this iOS 13 SSO extension? Well, not all IDP provider are skilled enough to provide a public Kerberos realm, this extension will so let limited IDP to provide a better solution. But this will require an additional app for that, and so the UX wont be as good as Kerberos, since the installation order of managed app can’t be guaranteed

SSO on macOS

Working since macOS 10.10, better since 10.12

Based on certificate based authentication

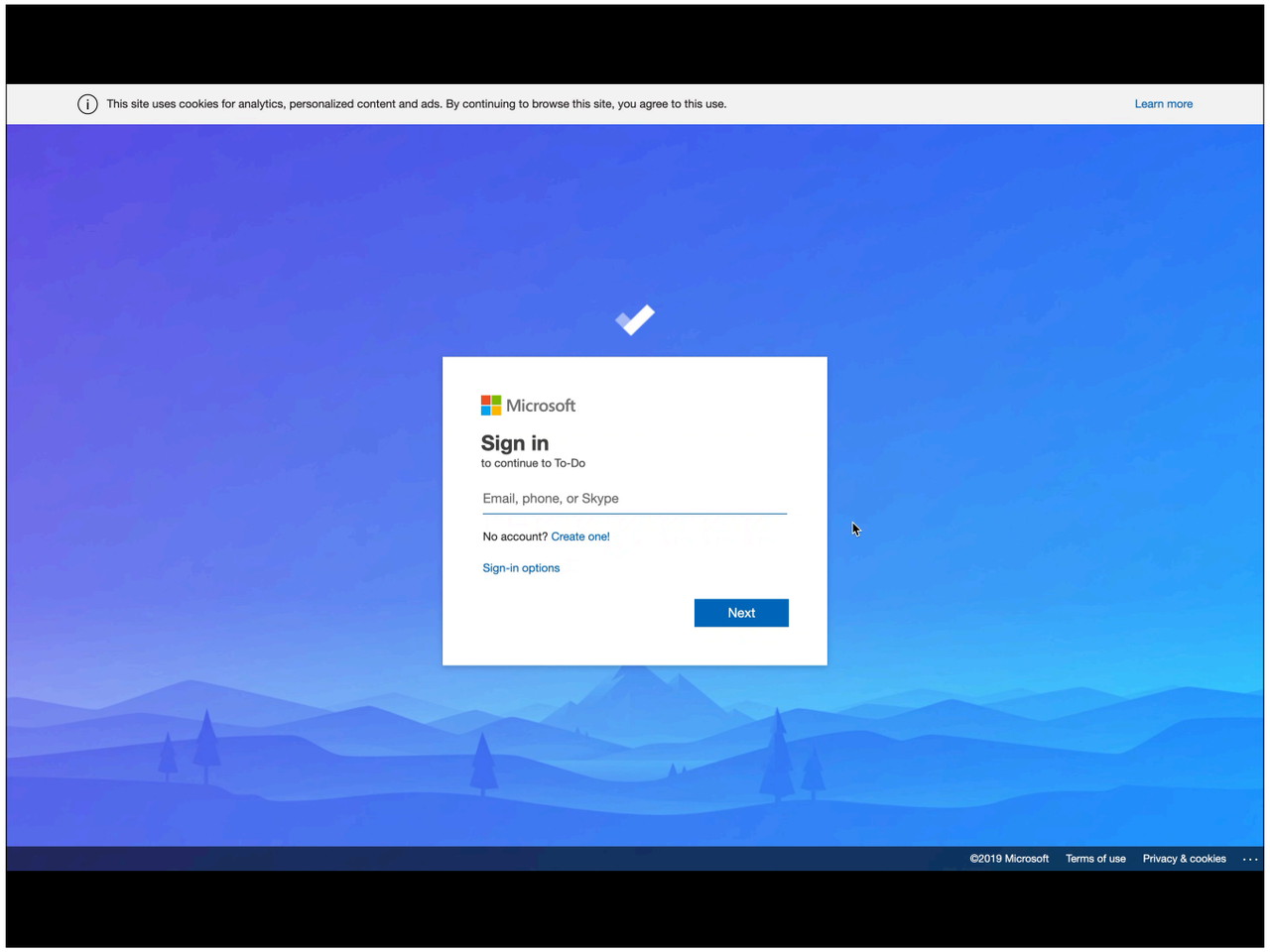
Support also Kerberos

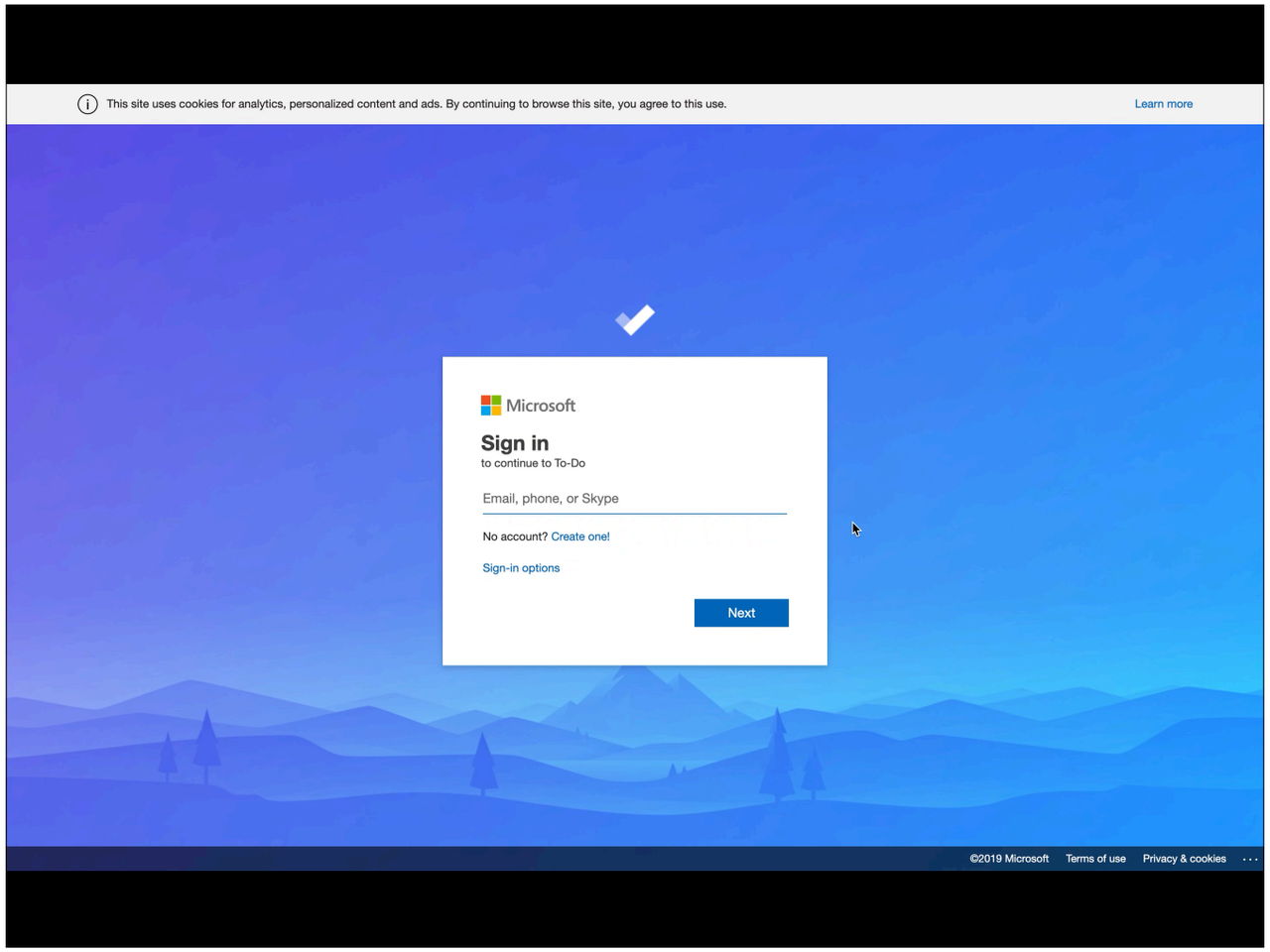
Fully provided by macOS, small developer work needed

For macOS, SSO can rely on Kerberos too, but it’s more easy to use Certificates based authentication most of the time, it’s better for mobility. And it work since 10.10 and got better in 10.12

Everything is handled by macOS but developer has to read and follow the documentation. So it can break if the developer is bad…

Here is how it look like




This site uses cookies for analytics, personalized content and ads. By continuing to browse this site, you agree to this use.
Learn more
Microsoft
Sign in
to continue to To-Do
Email, phone, or Skype
No account? Create one!
Sign-in options
Next
©2019 Microsoft
Terms of use
Privacy & cookies
...

macOS 10.15 SSO extension?

Not all IDP are good enough to handle certificates

Should extend the capabilities if your IDP is limited

Will require additional app

UX not as good if extension get installed last

Should support mobile Kerberos like iOS

macOS Catalina will get SSO extension too, and for the same reasons, not all IDP on the market are able to handle certificates, and so this option will provide more way for limited IDP to do a better job, and the same limits apply for the UX.

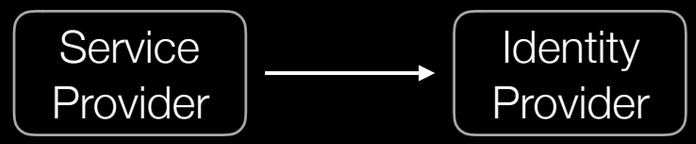
But since macOS Catalina will now have native integration for Enterprise Connect, we may be able to use the Mobile Kerberos currently used for iOS with it

IDP Chaining

And there is a last thing I want to talk about, IDP chaining. You have one SP requesting one IDP to handle authentication

IDP Chaining

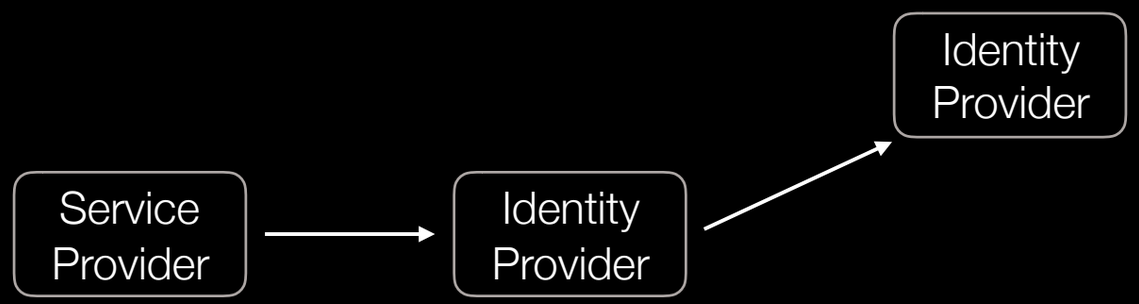
IDP Chaining



But an IDP could also rely on another IDP for some reasons

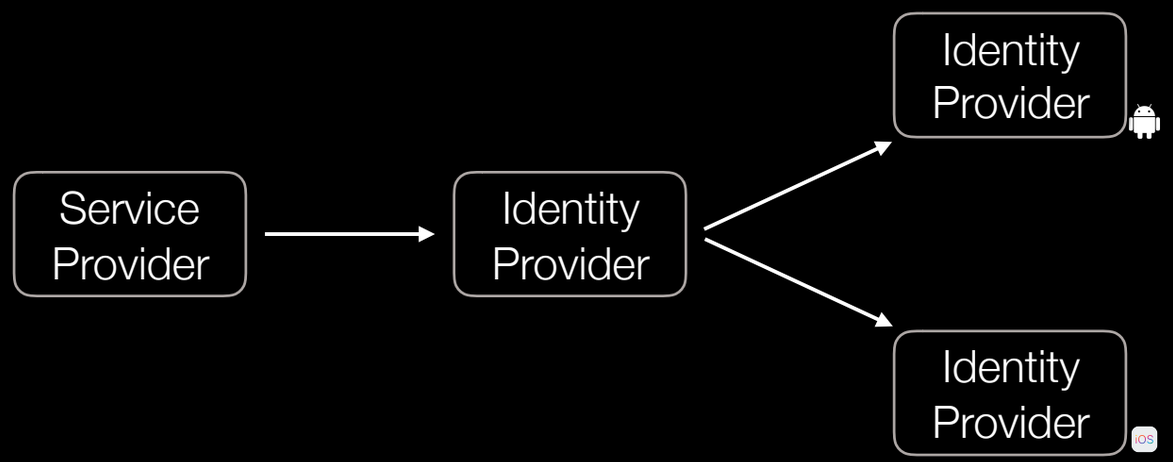
IDP Chaining

IDP Chaining



And eventually can do some logic here to request specialized IDP per requesting OS

IDP Chaining



That’s all for the day, I will speak again Friday morning about full cloud setup with Azure AD and Google Cloud Identity and how to use it with MDM

Subject	Room	Date
Going full cloud with Azure AD or Google Cloud Identity	207	Friday, 10:15am

Related talks

We now have time for Q&A
Thanks for attending this talk!

Thank you !

https://bit.ly/psumac2019-340