POV AGREEMENT

This Proof of Value Agreement ("Agreement") contains the exclusive terms and conditions between Cloudentity, Inc, a Delaware company, ("Cloudentity" or "Vendor") and AppDynamics LLC, a Delaware limited liability company ("AppDynamics"), and it governs AppDynamics testing, evaluation and use of, the software provided by Vendor (the "Software"). By accessing the Software described herein and signing below, the parties are consenting to be bound by and are becoming a party to the terms and conditions of this Agreement.

Evaluation; Term. For a period of 365 days (the "**Term**"), AppDynamics may download and/or access the Software and test the Software's functionality and look and feel, but only to test and evaluate the Software for its intended purpose internally within AppDynamics' organization. AppDynamics will not rent, sell, lease or otherwise transfer or allow access to the Software or any part thereof or use it for the benefit of a third party. AppDynamics will not reverse assemble, reverse compile or reverse engineer the Software, or otherwise attempt to discover any such Software source code or underlying Confidential Information (as defined below). This agreement will commence on 11/19/2020 (the "**Effective Date**") and will continue for the Term. Sections 2 and 4-5 of this Agreement will survive any expiration or termination of this Agreement.

Confidentiality. "**Confidential Information**" means all information of a party ("**Disclosing Party**") disclosed to the other party ("**Receiving Party**") that is designated in writing or identified as confidential at the time of disclosure or should be reasonably known by the Receiving Party to be confidential due to the nature of the information disclosed and the circumstances surrounding the disclosure. The terms of this Agreement, the Software, any technical or other documentation relating to the Software, logins, passwords and other access codes and any and all information regarding Vendor's business, products and services are Confidential Information. The Receiving Party will: (i) not use the Disclosing Party's Confidential Information for any purpose outside of this Agreement; (ii) not disclose such Confidential Information to any person or entity, other than its affiliates, employees, consultants, agents and professional advisers who have a "need to know" for the Receiving Party to exercise its rights or perform its obligations hereunder, provided that such employees, consultants, and agents are bound by agreements or, in the case of professional advisers, ethical duties respecting such Confidential Information in accordance with the terms of this section 2; and (iii) use reasonable measures to protect the confidentiality of such Confidential Information. If the Receiving Party is required by applicable law or court order to make any disclosure of such Confidential Information, it will first give written notice of such requirement to the Disclosing Party, and, to the extent within its control, permit the Disclosing Party to intervene in any relevant proceedings to protect its interests in its Confidential Information, and provide full cooperation to the Disclosing Party in seeking to obtain such protection. Further, this section 2 will not apply to information that the Receiving Party can document: (i) was rightfully in its possession or known to it prior to receipt without any restriction on its disclosure; (ii) is or has become public knowledge or publicly available through no fault of the Receiving Party; (iii) is rightfully obtained by the Receiving Party from a third party without breach of any confidentiality obligation; or (iv) is independently developed by employees of the Receiving Party who had no access to such information.

WARRANTY DISCLAIMER. The parties acknowledge that the Software is provided "AS IS." VENDOR DISCLAIMS ALL WARRANTIES RELATING TO THE SOFTWARE, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT.

Limitation of Remedies and Damages. NEITHER PARTY WILL BE RESPONSIBLE OR LIABLE WITH RESPECT TO ANY SUBJECT MATTER OF THIS AGREEMENT OR TERMS AND CONDITIONS RELATED THERETO UNDER ANY CONTRACT, NEGLIGENCE, STRICT LIABILITY OR OTHER THEORY (A) FOR LOSS OR INACCURACY OF DATA OR, COST OF PROCUREMENT OF SUBSTITUTE GOODS, SOFTWARE OR TECHNOLOGY, OR FOR ANY INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES INCLUDING, BUT NOT LIMITED TO LOSS OF REVENUES AND LOSS OF PROFITS, OR (B) ANY OTHER AMOUNTS THAT EXCEED $5,000 USD.

Equitable Relief; Miscellaneous. The parties acknowledge and agree that due to the unique nature of the Disclosing Party's Confidential Information, there can be no adequate remedy at law for any breach of the Receiving Party's obligations hereunder, that any such breach may allow the Receiving Party or third parties to unfairly compete with the Disclosing Party resulting in irreparable harm to the Disclosing Party, and therefore, that upon any such breach or threat thereof, the Disclosing Party will be entitled to injunctions and other appropriate equitable relief in addition to whatever remedies it may have at law, without any obligation to post a bond. In the event that any of the provisions of this Agreement will be held by a court or other tribunal of competent jurisdiction to be unenforceable, such provisions will be limited or eliminated to the minimum extent necessary so that this Agreement will otherwise remain in full force and effect and enforceable. This Agreement constitutes the entire agreement between the parties pertaining to the subject matter hereof, and any and all written or oral agreements previously existing between the parties are expressly cancelled. Neither the rights nor the obligations arising under this Agreement are assignable or

transferable by the other party, and any such attempted assignment or transfer will be void and without effect. This Agreement will be governed by and construed in accordance with the laws of the State of California without regard to the conflicts of laws provisions therein.

IN WITNESS WHEREOF, the parties to this Agreement have caused this Agreement to be executed by their authorized representatives as of the date of the last signature below.

AppDynamics LLC

DocuSigned by:

By: George Karamanos
EADD3AC1FF6B486...

Printed Name: George Karamanos

Title: General Counsel

Date: November 24, 2020

Vendor: Cloudentity, Inc.

DocuSigned by:

By: Jasen Meece
B826D824F5DB47D...

Printed Name: Jasen Meece

Title: CEO

Date: November 24, 2020

AuthZ PoC

Purpose

To evaluate whether provided software capabilities match AppD use cases, including a portability of existing legacy RBAC solution and the future requirements which are not currently supported in the existing AppD solution.

Duration

PoC should be completed within two weeks.

Success criteria

All use cases below can be satisfied with existing software capabilities or require a minimal additional effort.

Test framework

Identities

Working assumption is that the tokens are granted in the context of a specific tenant.

Two types of identities, each is a JWT token that matches a specific condition.

- User
  - condition: idType='user'
  - claims: sub, accountId, tenantId, groups, email
- App
  - condition: idType='app'
  - claims: sub, accountId, tenantId

Graph

Graph should be hosted in AWS Neptune and accessed via Gremlin: https://docs.aws.amazon.com/neptune/latest/userguide/access-graph-gremlin.html

A sample graph will be provided either as a snapshot or a set of executable Gemlin queries.

Organization

AppDynamics is a resource owner, selling a product to Pepsi and Coca-Cola:

- AppDynamics
  - Pepsi (accountId=pepsi)
    - Pepsi US (tenantId=pepsi_us)
      - Marketing
      - Sales
    - Pepsi EU (tenantId=pepsi_eu)
      - Marketing
      - Sales
  - Coca-Cola (accountId=cocacola)
    - Coca-Cola US (tenantId=cocacola_us)
      - Marketing
      - Sales
    - Coca-Cola EU (tenantId=cocacola_eu)
      - Engineering
      - Sales

Asset types

There are two types of assets in AppD: graph aware and regular. For graph aware assets, we need to support two use cases:

1. a graph unaware service should be able to evaluate a policy and get back a list of entity identifiers to scope the data for: this involves a call from PDP to the defined PIP (our graph backend via REST API), which evaluates this condition and returns back a set of ids to pass to the next service
2. a graph aware service should be able to ask a question to PDP: what are the conditions under which a given identity is allowed an access to this resource? This should return an expression which represents combined conditions defined in all policies applicable to this asset type and identity

Sample asset definitions:

- entity (graph aware)
  - access types: update_entity, view_entity, write_metric, view_metric
  - attributes:
    - tags: array<string>
    - tenantId: string
    - name: string
    - type: string
    - children: array<entity> - represents entities connected via outgoing edges in a graph
- dashboard
  - access types: update, view
  - attributes:
    - tags: array<string>
    - tenantId: string
    - ownerId: string
    - name: string

Deployment models for PEP

- sidecar - for microservices running in k8s
  - how is the mapping from the incoming request to the resource and the action configured?
- service calling PDP directly - for services outside of k8s

Use cases

- Management:
  - As an AppD product owner, I want to manage my asset definitions and the access that they support: onboarding, updating, deprecating, retiring
  - As an AppD product owner, I want to evaluate test policies against my product
- Administration:
  - As an AppD administrator, I want to be able to configure a set of virtual identity types (based on claims in JWT tokens)
  - As an AppD administrator, I want to be able to configure policies at a global level that are enforced in all tenants
    - example: ensure that identities logged into a specific tenant can only view/manage the assets of that tenant
  - As an AppD administrator, I want to configure multiple tenants with a specific set of identities
    - example: ensure that only users from a specific account can be delegated for this tenant administration
  - As an AppD administrator, I want to be able to delegate administration to specific user(s) in the above set
- Delegated administration:
  - As a Customer administrator, I want to configure multiple workspaces with a specific set of identities
  - As a Customer administrator, I want to be able to manage my own policies, and (optionally) apply them across workspaces
  - As a Customer administrator, I want to be able to configure policy approval workflows
  - As a Policy approver, I want to be able to approve/reject access requests
  - As a Customer administrator, I want to restrict access to identities within a specific IP range, so that I can ensure no outside traffic accesses my org's data
  - As a Customer administrator, I want to simulate what the user's access will look like in the product UI, so that I can validate they have the necessary access to carry out a task
  - As a Customer administrator, I want to audit who created/updated/approved/deleted a policy, so that I can continue to have proper monitoring controls in place
  - As a Customer administrator, I want to provide access to limited assets for a specific time period, so that users can have the necessary access to complete root-cause analysis or any other time sensitive task
- End user:
  - As an end-user, I want to request access to an asset and have an approval workflow sent to the necessary person, so that I don't have to leave the product to make the same request.
- Policy execution:
  - As a UI client, I want to be able to find out capabilities of a specific user
  - As a background app, I want to be able to find out under which condition a specific identity can perform an operation on a resource
- Policy examples:
  - Entities:
    - Users in Admins group can only view metrics for entities of type 'service' if they belong to a subgraph under an entity with a tag 'env:prod'
    - Users in Observer group cannot view metrics for any entities that belong to a subgraph under an entity with a tag 'env:prod'
    - Apps from account pepsi can only write metrics for entities with type='app' and name='pepsi app' and any of their children
  - Dashboards:
    - Users can only manage dashboards if they own them
    - Users can grant view access to the dashboards they own

Integration

PoC should be deployed and managed by the company in their AWS account, the environment should be made accessible to the following set of AppD employees: Anna Bokhan Jay Desai Fabian Gonzales