AUTHORIZATION

ENABLING BUSINESS

TECHVISION
CHRYSALIS

Authorization Challenges

Faster service delivery

Modernize apps

Authorization volume, breadth

Multi-cloud workload complexity

Compliance

Increased Cyberattacks

What business are facing…

Delayed application delivery
Prolonged security verification effort and time due to varying access and authorization controls

Hard Coded
authorization entitlements and consent governance are embedded

Authorization is inconsistent and inadequate
- Fragmented, hard-coded, per application AuthZ
- Can't understand and invoke common policy across apps, services, APIs
- Course-grained authorization policy and enforcement limitations

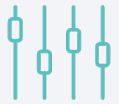
Development and DevSecOps inefficiency
- Piecemeal API/service visibility and on-boarding
- Decentralized authorization management and provisioning complexity

Increased attack, privacy, audit and compliance exposure
App/service/API access, data and audit blind spots

TECHVISION CHRYSALIS®

The Desired State

Digital business acceleration



Authorization governance

Increased DevOps proficiency

Threat and compliance risk mitigation

Authorization Requirements

Externalized Authorization & Governance

- Decouple from apps/APIs and add versioning/governance
- Streamline creating fine-grained authorization policy as code
- Standardize policies to invoke within legacy/new apps, services and APIs

Enable Developers

- Rapid user, app and API onboarding
- Integrate privacy workflow and data protection across apps, services, APIs

Zero Trust

- Granular control on all service/API ingress and egress decision points
- Prevent unauthorized N/S perimeter and E/W lateral access and data leakage

Scalability

- Dynamically enforce policy at hyperscale for every transaction
- Invoke native control as close to the service/app/API ie. Kubernetes

TECHVISION
CHRYSALIS®

Types of Authorization

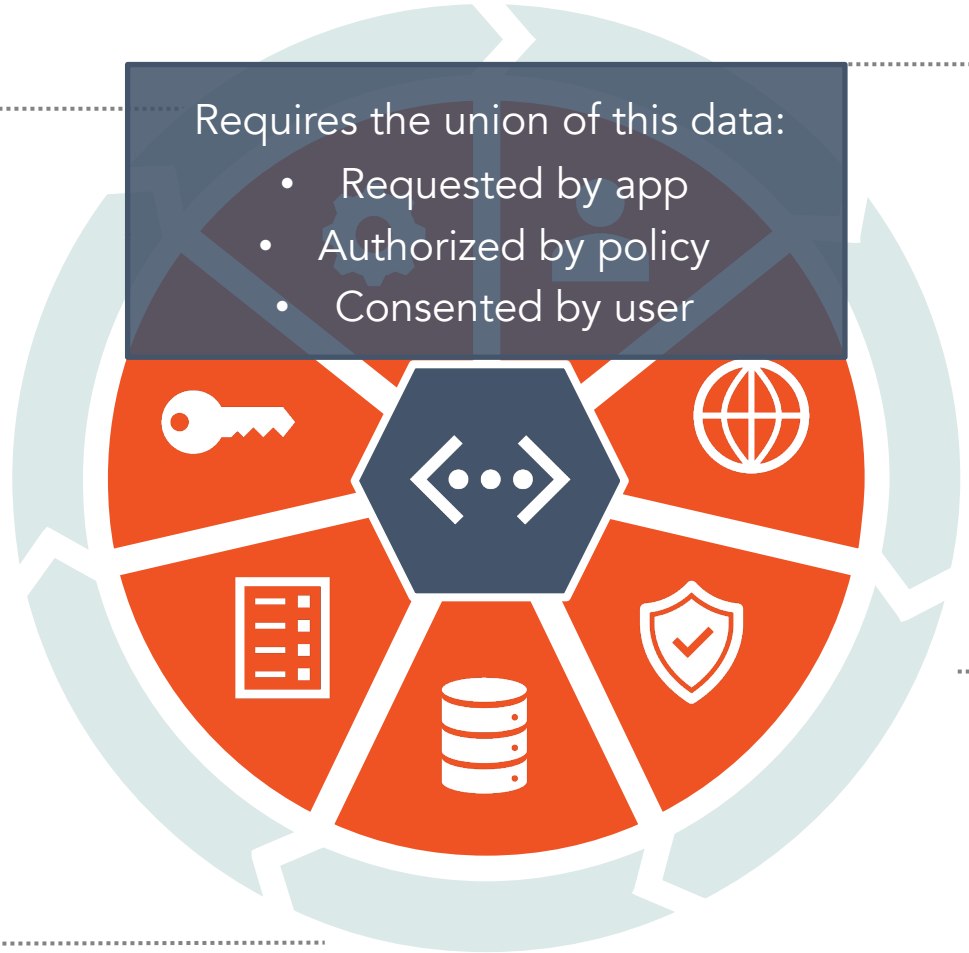
- Coarse Grained
  - RBAC -- Group or Role Based. “Is user a citizen?”
  - ABAC -- Attribute based “is user over age 12?
- Mid Grained
  - Relationship Based “Is user a family member?”
  - Mid-Grained permissions “Is user a licensee?”
- Fine-grained
  - “Can user execute region specific JavaScript within SPA?”
- Privacy
  - “Did user consent to sharing data x?”

Customer Business Problem

I want to allow minors to manage top 10 game orders for other family members within an application while following regional regulations with full auditability

Requires the union of this data:

- Requested by app
- Authorized by policy
- Consented by user



Access Requirements

How did the User Authenticate?

Can user access application?

Consent Granted

Did user consent to sharing last name and favorite games?

Regulations Applied

Data Residency
COPA- Is User over the age of12?

User Identity

What profile data and from where?

Relationship Graph

Is user member of family?

Risk & Fraud Intelligence

Is User accessing from his registered countries IP range?

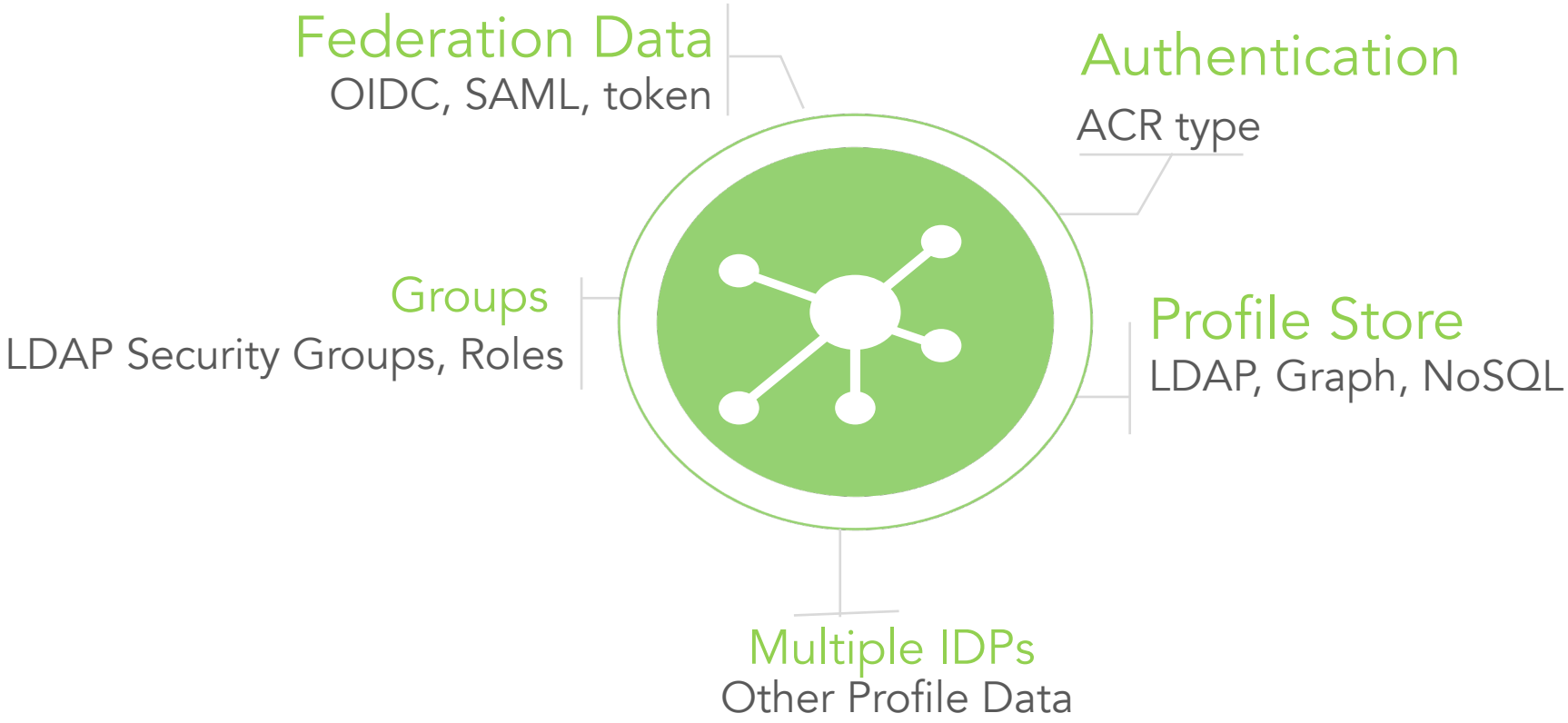
What is Users Risk?

Action Requested

Is user in role manager? Was role granted by parents

TECHVISION
CHRYSALIS®

User Context



Authorization Context

App Entitlements
Permissions, transactional value, service identity

RBAC PBAC ABAC
Entity, group/role, authentication, geo attributes

Permissions and Relationship
Zanzibar store for relationship management

External Metadata
Time of day, WAF data

Consent
Did the User allow the transaction

Authorized
User, service, API transaction granted or trigger step-up in authentication parameters

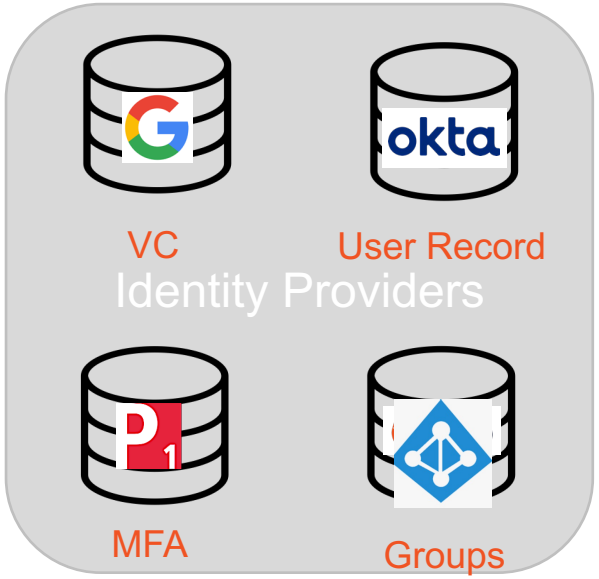
Who? What? Where? Why? When?

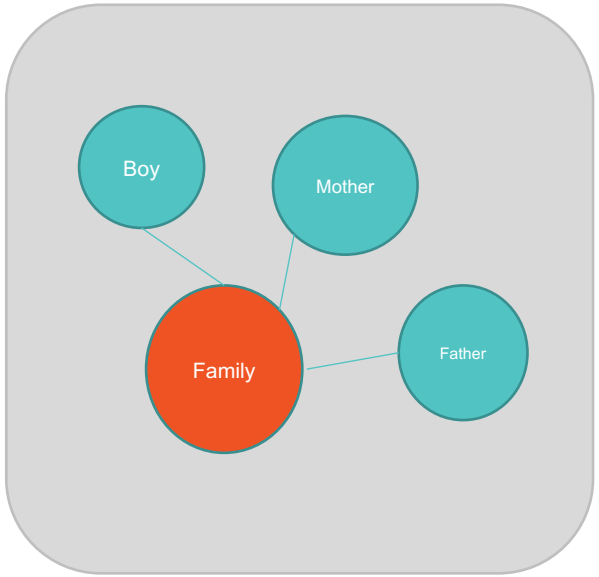
TECHVISION CHRYSALIS®

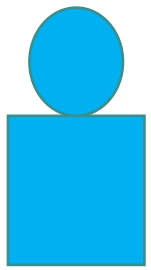
Policies OPA/REGO

- Decouples Policy Management and Enforcement from the application
- Increased Visibility of Policies, Versioning & Management
- Declarative Developer Friendly Language
  - Authorization as code
  - PiggyBack on OAuth
- External Data
  - Queries in real time
- Built in proxying and caching

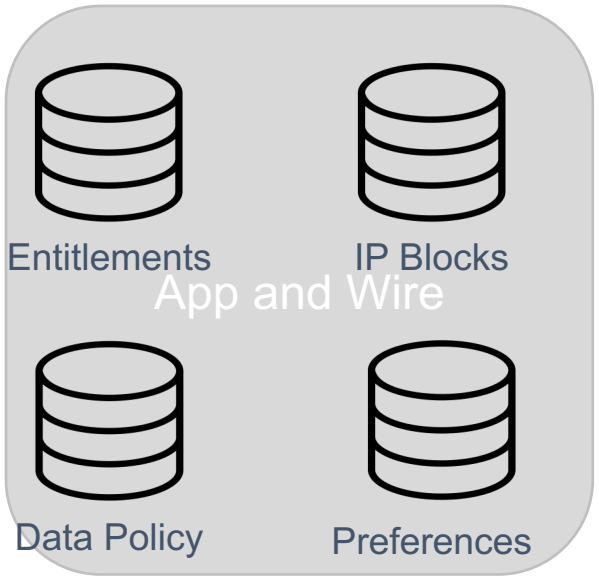
- OpenSource and CNCF “Graduated”

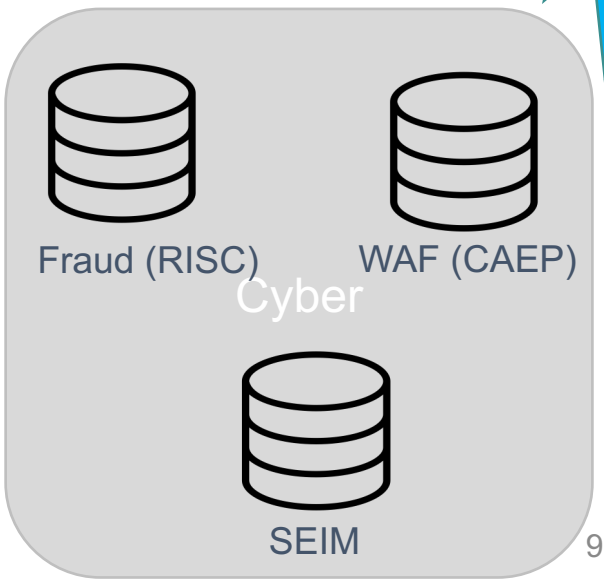






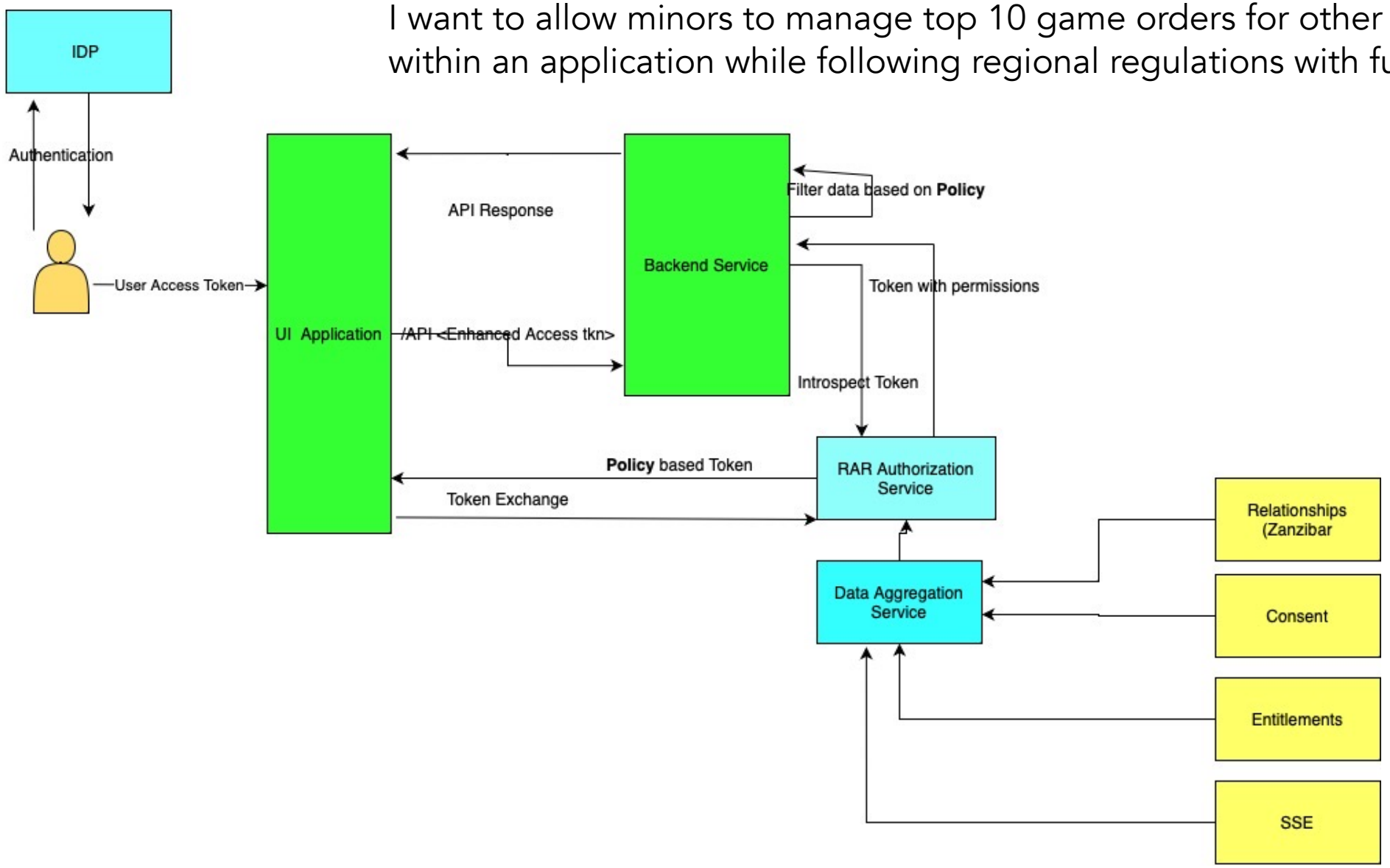
Identity → Coarse Grained → Entitlements → Fine-Grained → Relations → Cyber → JITA





Integration Pattern: OAuth w/RAR

I want to allow minors to manage top 10 game orders for other family members within an application while following regional regulations with full auditability



OPA Example

```
package envoy.http.jwt

import future.keywords.if

default allow := false

allow if {
                    is_post
                    is_family
                    action_is_update
                    claims.username == "alice"
}

is_post if input.attributes.request.http.method == "POST"

is_family if input.attributes.request.http.path == "/API/app_mgt"

claims := payload if {
                    # Verify the signature on the Bearer token..
                    io.jwt.verify_hs256(bearer_token, "B41BD5F462719C6D6118E673A2389")

                                        # In Rego, you can pattern match values using the `=` and `:=` operators. This
                    # example pattern matches on the result to obtain the JWT payload.
                    [_, payload, _] := io.jwt.decode(bearer_token)
}

bearer_token := t if {
                    # Bearer tokens are contained inside of the HTTP Authorization header. This rule
                                        v := input.attributes.request.http.headers.authorization
                    startswith(v, "Bearer ")
                    t := substring(v, count("Bearer "), -1)
}



                    }

user_is_owner if data.user_attributes[input.user].title == "owner"

user_is_family if data.user_attributes[input.user].title == ”son"

user_is_senior if data.user_attributes[input.user].tenure > 12
```


```
"request": {
          "http": {
             "headers": {
                ":authority": "example-app",
                ":method": "POST",
                ":path": "/API",
                "accept": "*/*",
                "authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW
                "content-length": "0",
                "user-agent": "curl/7.68.0-DEV",
                "x-ext-auth-allow": "yes",
                "x-forwarded-proto": "http",
                "x-request-id": "1455bbb0-0623-4810-a2c6-df73ffd8863a"
             },
             "host": "example-app",
             "id": "8306787481883314548",
             "method": "POST",
             "path": "/API/app_mgt",
             "protocol": "HTTP/1.1"
          }
       },

    },
    ""authorizationDetails":
{ "elements": [
       { "actions": [
                          "read",
                          ”update”
                            ],
            "dataTypes": [
                          ”owner",
                          ”son"
                            ],
          ”locations": [
                           "https://example.com/location.list"
                            ], ”
                          type": "customer_profile"
          ”Age":
                          { ”Years Old": ”12",
                          }
           ]

},
```

Authorization solves business identity problems

- Stop Data leaks
  - Entitlements control what data can be used by what application
  - Normalize Data & Share only attributes required

- Support API-first and cloud-native development
  - Policy as Code simplifies developer adoption
  - Policy Versioning and Auditability
  - OpenStandards (Oauth & OIDC)

- Simplify Privacy
  - Consent can be checked on every data access
  - Pull real-time data from existing IDPs & data stores

- Scale with Automated integration of Modern Applications
  - Kubernetes, Functions, APIs and Service Meshes

TECHVISION

CHRYSALIS