The Future of Authorization in Applications

Gabriel L. Manor


AUTHORIZATION?
WHERE WE'RE GOING, WE DON'T NEED AUTHORIZATION
imgflip.com

Gabriel L. Manor

Engineering Director @ Permit.io

🏆 **Implementing Software Solution in His Perosnal Life**

Not an ethical hacker, zero awards winner, dark mode hater.

Chaos

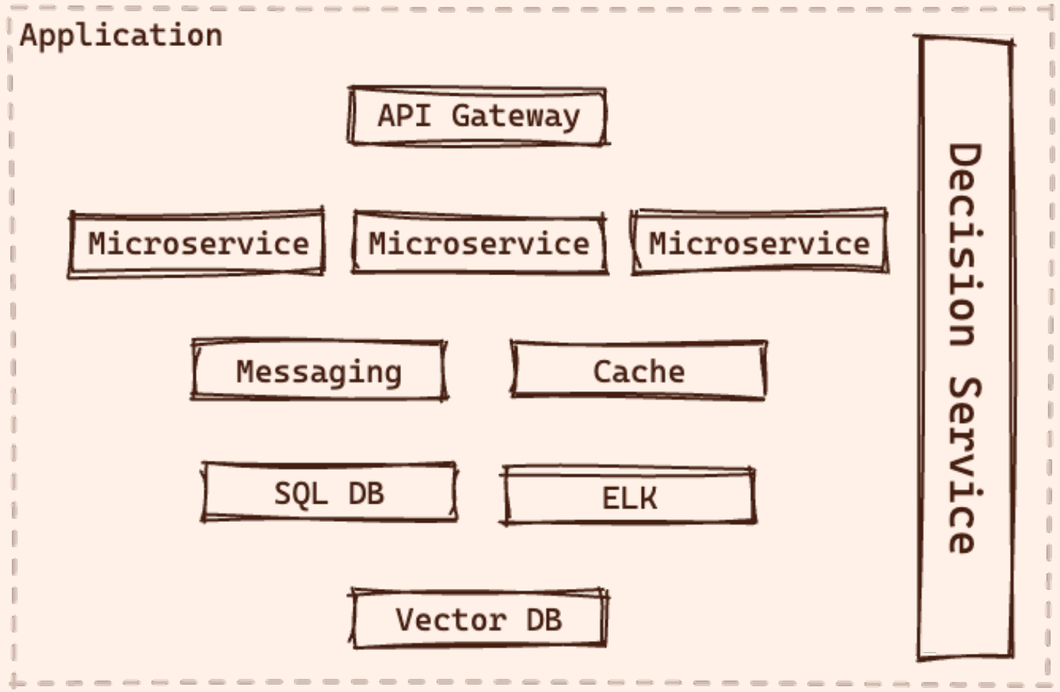
Complexity

Velocity

Confidence

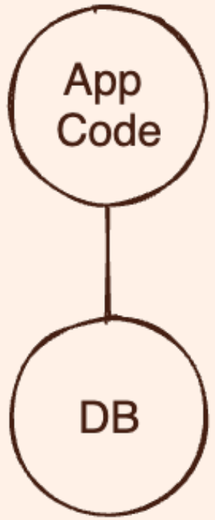
Chaos, Complexity, Velocity, Confidence

Monolith

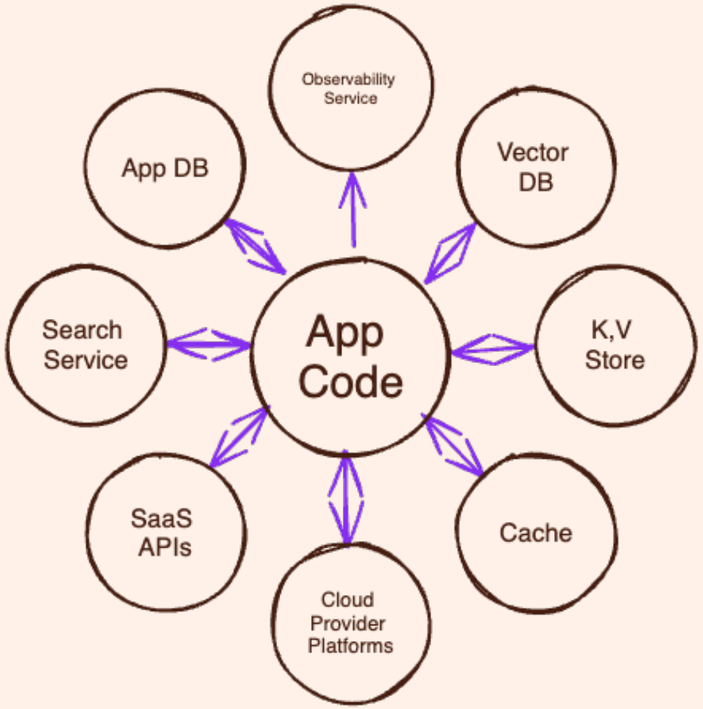


Cloud Native

Chaos, *Complexity*, Velocity, Confidence

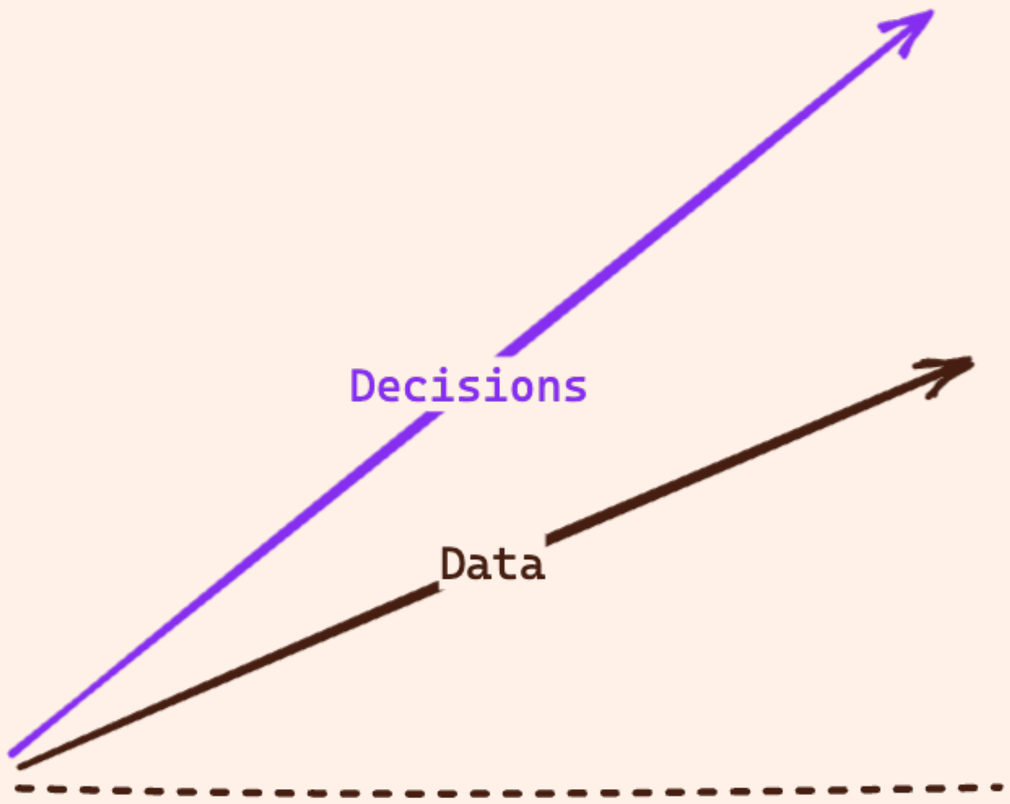


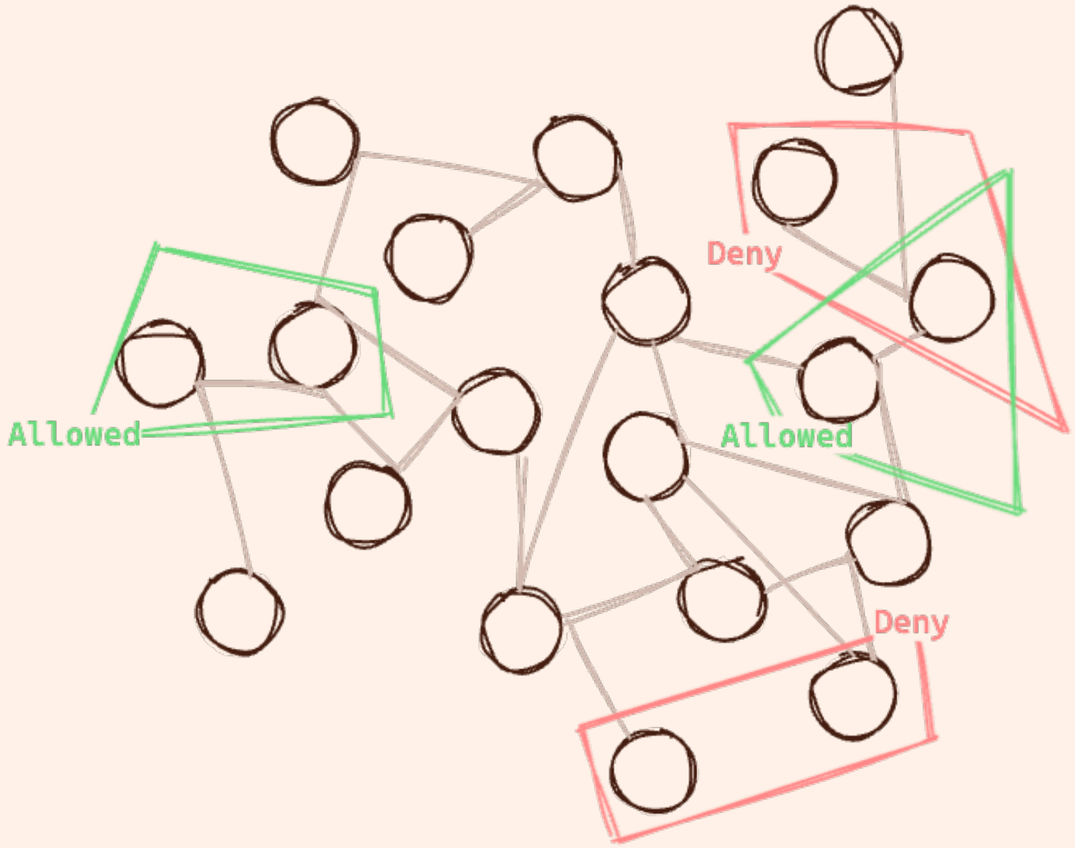
2010



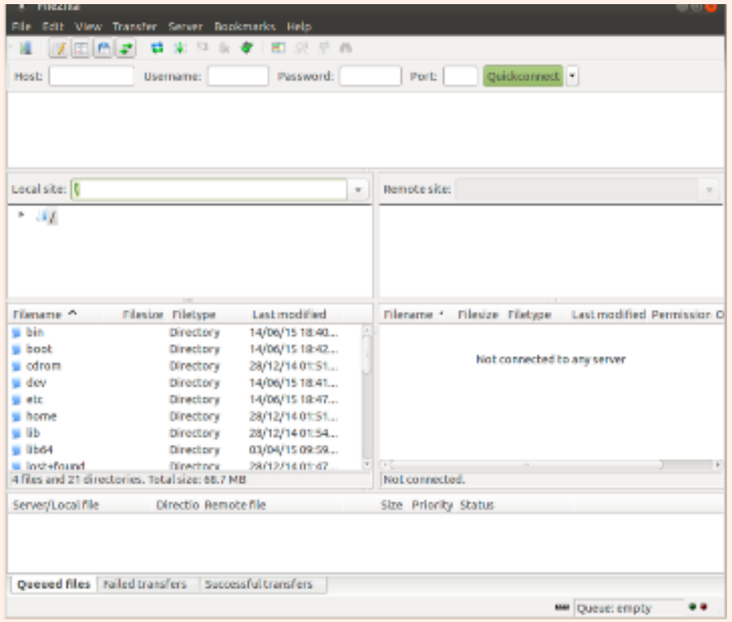
2024

Decision Fatigue Syndrome

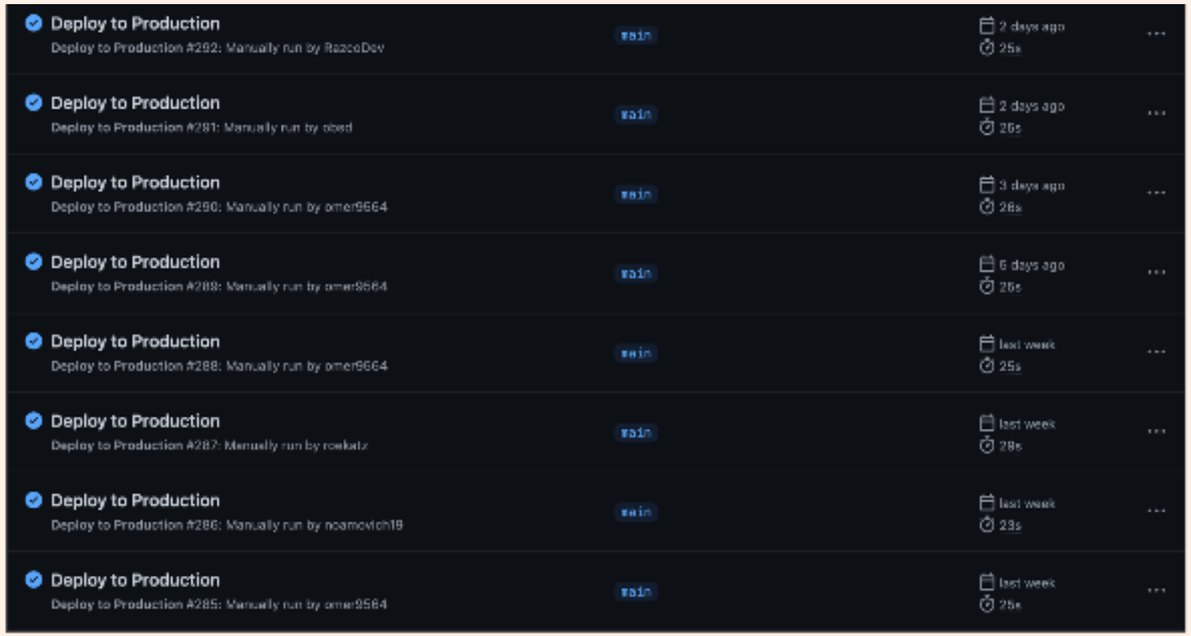




Chaos, Complexity, *Velocity*, Confidence



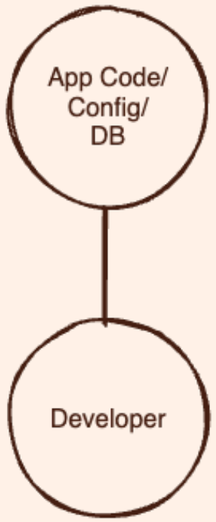
Monthly Night Work



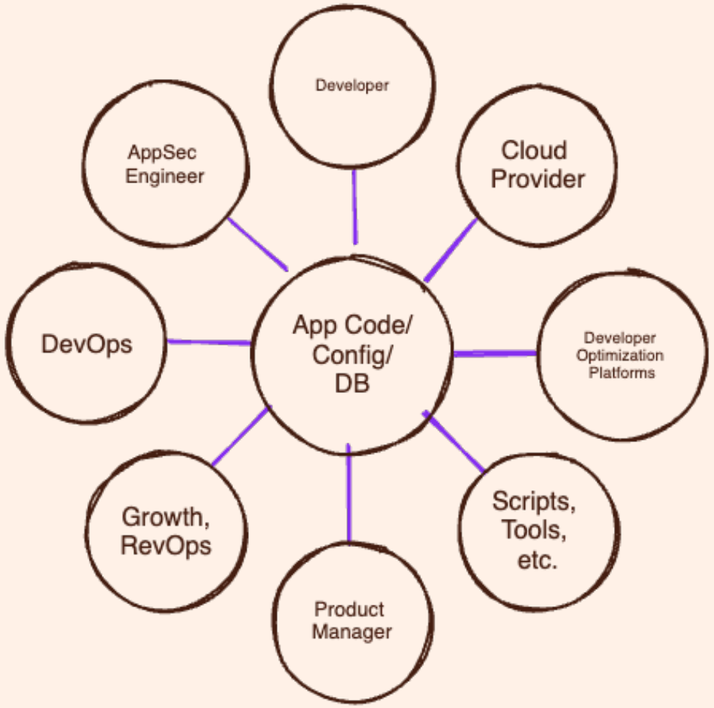
Multiple Daily Deployments

Chaos, Complexity, Velocity, *Confidence*

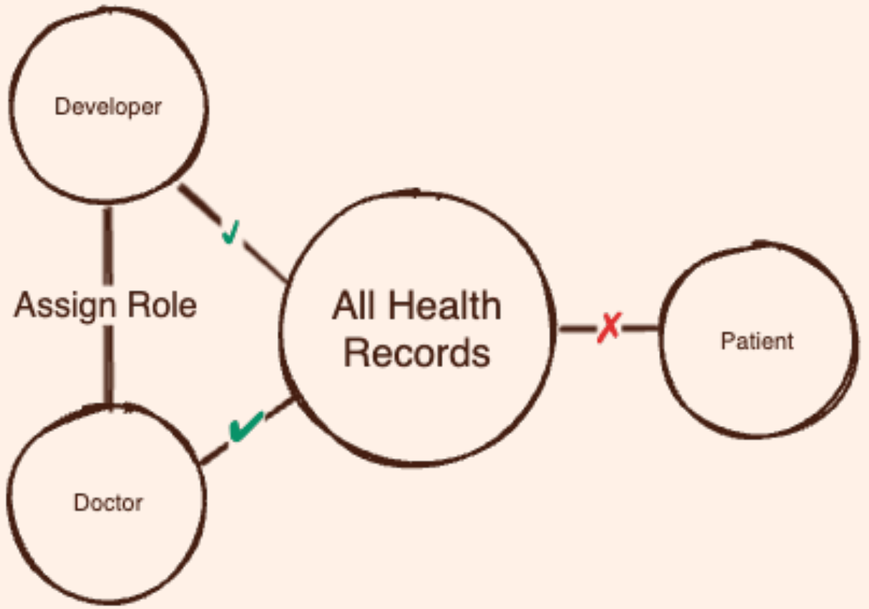
Application Code Stakeholders



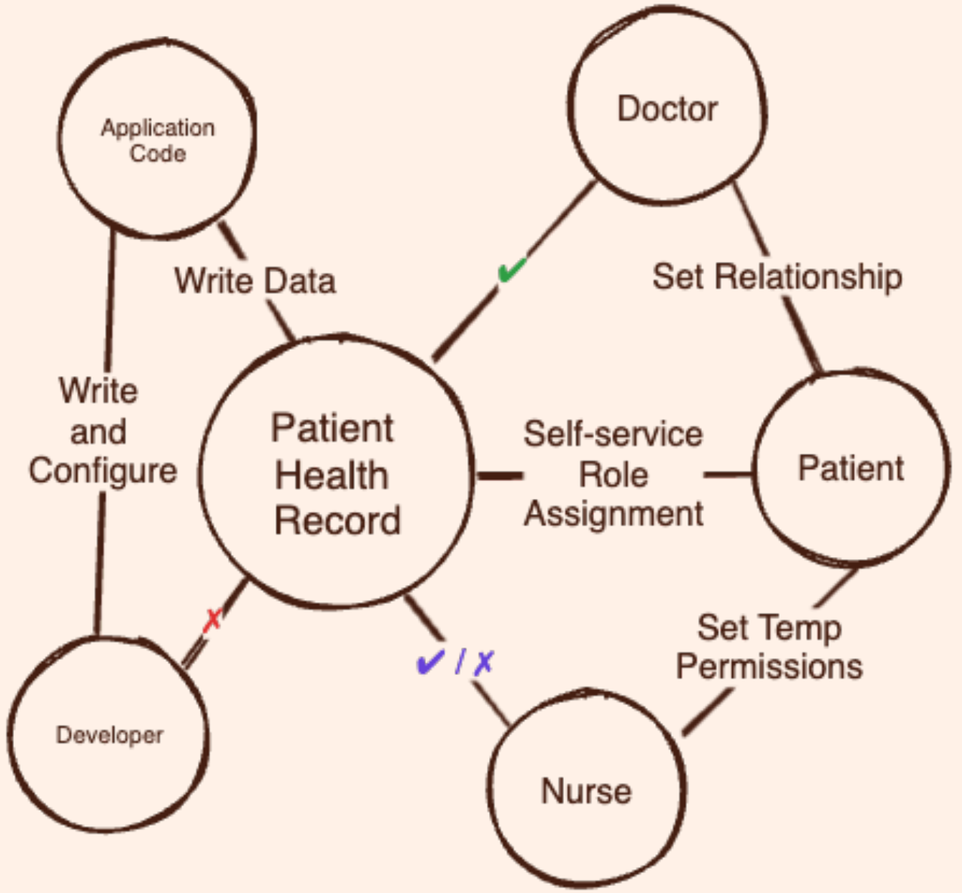
2010



2024


Developer
Assign Role
Doctor
All Health Records
Patient

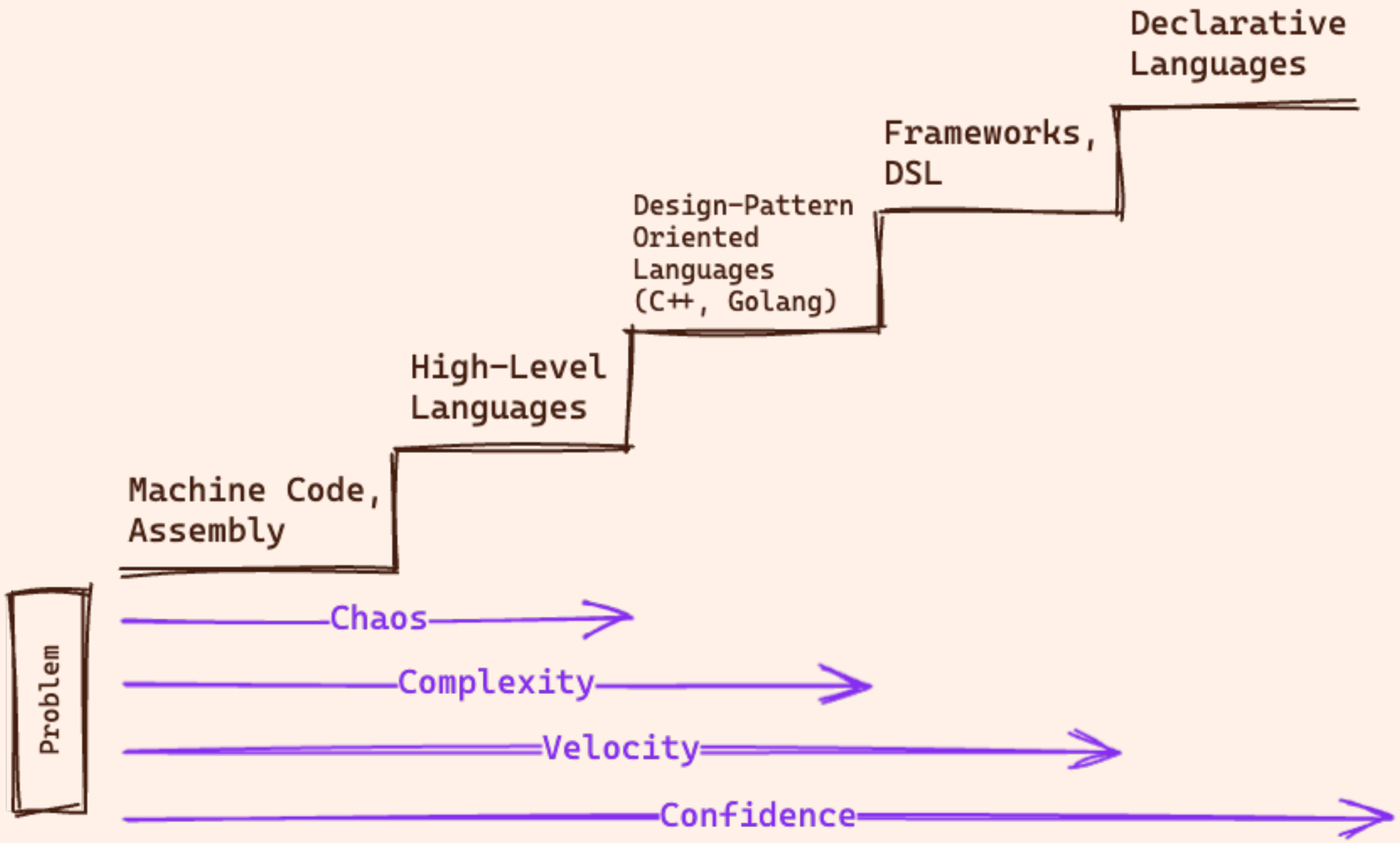
2018


Application Code
Write Data
Write and Configure
Developer
Doctor
Set Relationship
Patient Health Record
Self-service Role Assignment
Patient
Set Temp Permissions
Nurse

2024


AI AGENTS, LLM
LIGHTYEAR
THEY ARE EVERYWHERE
imgflip.com


PRODUCT
ANOTHER PRODUCT
MY PRODUCT
AUTHORIZATION
imgflip.com
memegenerator.net


Declarative Languages
Frameworks, DSL
Design-Pattern Oriented Languages (C++, Golang)
High-Level Languages
Machine Code, Assembly
Problem
Chaos
Complexity
Velocity
Confidence


Kids, violence is never the answer
HTML is a Programming Language

HTML

```html
<div>This is a div in HTML.</div>
```

C++

```cpp
#include <windows.h>

// Function declarations
LRESULT CALLBACK WindowProcedure(HWND, UINT, WPARAM, LPARAM);
void DrawRectangle(HWND hwnd);

// Entry point
int WINAPI WinMain() {
    WNDCLASSW wc = {0};

    wc.hbrBackground = (HBRUSH)COLOR_WINDOW;
    wc.hCursor = LoadCursor(NULL, IDC_ARROW);
    wc.hInstance = hInst;
    wc.lpszClassName = L"myWindowClass";
    wc.lpfnWndProc = WindowProcedure;

    if (!RegisterClassW(&wc)) return -1;

    CreateWindowW(
        L"myWindowClass",
        L"Simple Div in C++",
        WS_OVERLAPPEDWINDOW | WS_VISIBLE,
        100,
        100,
        500,
        500,
        NULL,
        NULL,
        NULL,
        NULL
    );

    MSG msg = {0};
    while (GetMessage(&msg, NULL, 0, 0)) {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }

    return 0;
}

// Window procedure
LRESULT CALLBACK WindowProcedure(HWND hwnd, UINT msg, WPARAM wp, LPARAM lp) {
    switch (msg) {
    case WM_DESTROY:
        PostQuitMessage(0);
        break;
    case WM_PAINT:
        DrawRectangle(hwnd);
        break;
    default:
        return DefWindowProcW(hwnd, msg, wp, lp);
    }
    return 0;
}

// Function to draw a rectangle
void DrawRectangle(HWND hwnd) {
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hwnd, &ps);
    // Set up the rectangle dimensions
    RECT rect = { 50, 50, 250, 150 };
    // Set up the brush color
    HBRUSH brush = CreateSolidBrush(RGB(0, 0, 255));
    FillRect(hdc, &rect, brush);
    EndPaint(hwnd, &ps);
}
```

Assembly

```asm
section .data
    className db 'WinClass',0
    windowName db 'Window',0

section .bss
    hInstance resb 6
    msg resb 28
    hwnd resb 4
    wndClass resb 48

section .text
    global _start

_start:
    ; Get instance handle
    call [GetModuleHandle]
    mov [hInstance], eax

    ; Define window class
    mov eax, [hInstance]
    mov [wndClass+36], eax
    mov dword [wndClass+40], WindowProc
    mov dword [wndClass+44], 0x0003
    mov [wndClass], 0x00000000
    lea eax, [className]
    mov [wndClass+5], eax
    call [RegisterClass]

    ; Create window
    push 0
    push 0
    push 0
    push 0
    push 0
    push 0
    push 0x3f0000
    push windowName
    push className
    push 0x00cf0000
    call [CreateWindowEx]
    mov [hwnd], eax

    ; Show and update window
    push 1
    push [hwnd]
    call [ShowWindow]
    push [hwnd]
    call [UpdateWindow]

    ; Message loop
    .message_loop:
        push 0
        push 0
        push [hwnd]
        lea eax, [msg]
        push eax
        call [GetMessage]
        test eax, eax
        jz .exit
        lea eax, [msg]
        push eax
        call [TranslateMessage]
        lea eax, [msg]
        push eax
        call [DispatchMessage]
        jmp .message_loop

    .exit:
        mov eax, 1
        xor ebx, ebx
        int 0x80

; Window procedure
WindowProc:
    push ebp
    mov ebp, esp
    cmp dword [ebp+12], 0x000F
    je .wm_paint
    cmp dword [ebp+12], 0x0002
    je .wm_destroy

    ; Default processing
    push dword [ebp+16]
    push dword [ebp+12]
    push dword [ebp+8]
    call [DefWindowProc]
    jmp .done

    .wm_paint:
        ; Handle WM_PAINT message here to draw the rectangle
        ; This part involves setting up a device context, drawing commands, etc.

        jmp .done

    .wm_destroy:
        push 0
        call [PostQuitMessage]
        xor eax, eax
        jmp .done

    .done:
        mov esp, ebp
        pop ebp
        ret
```

Domain-Specific Declarative Languages

SQL

YAML

LilyPond

CEL

XML

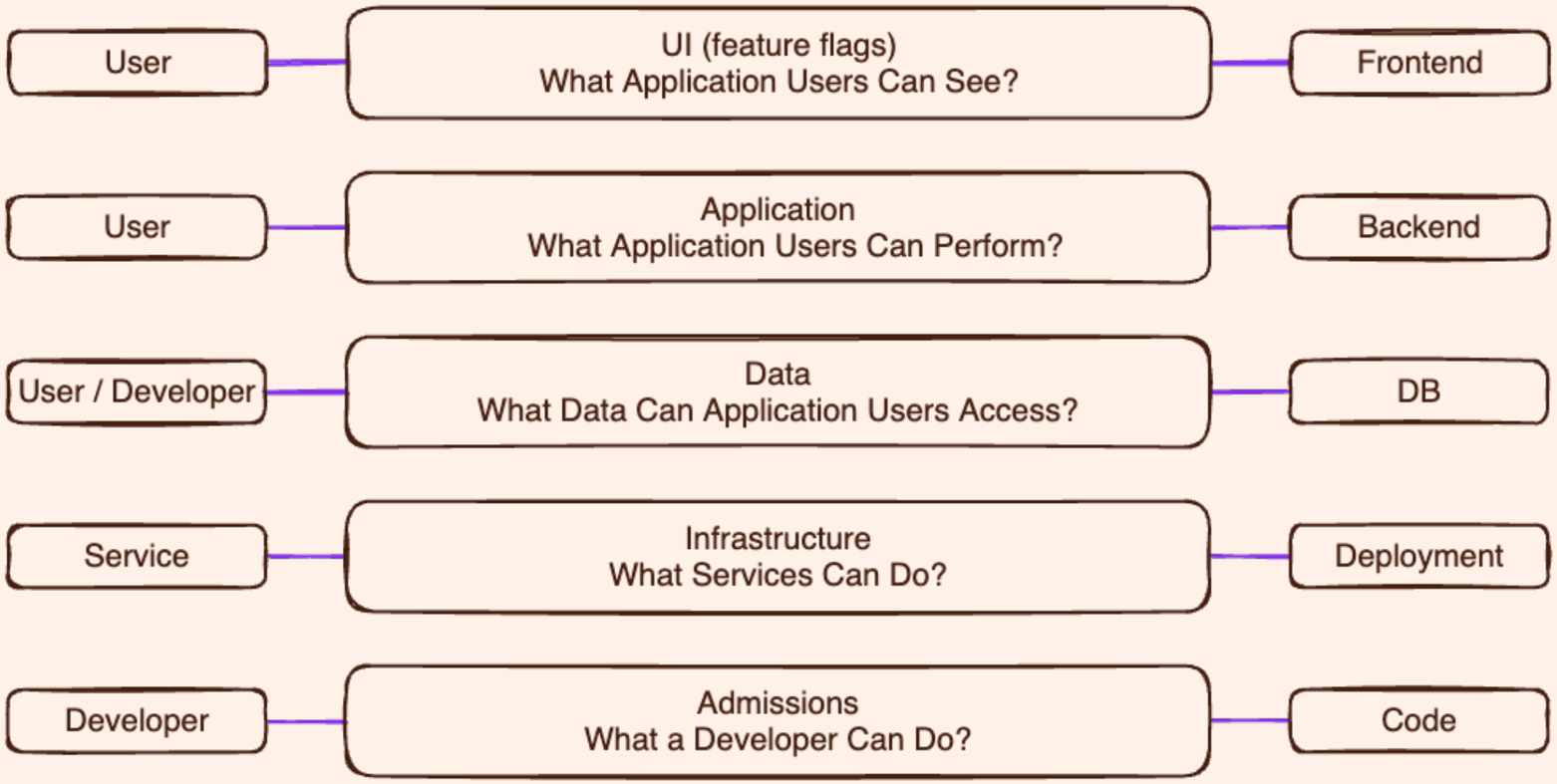
LaTeX

Gherkin

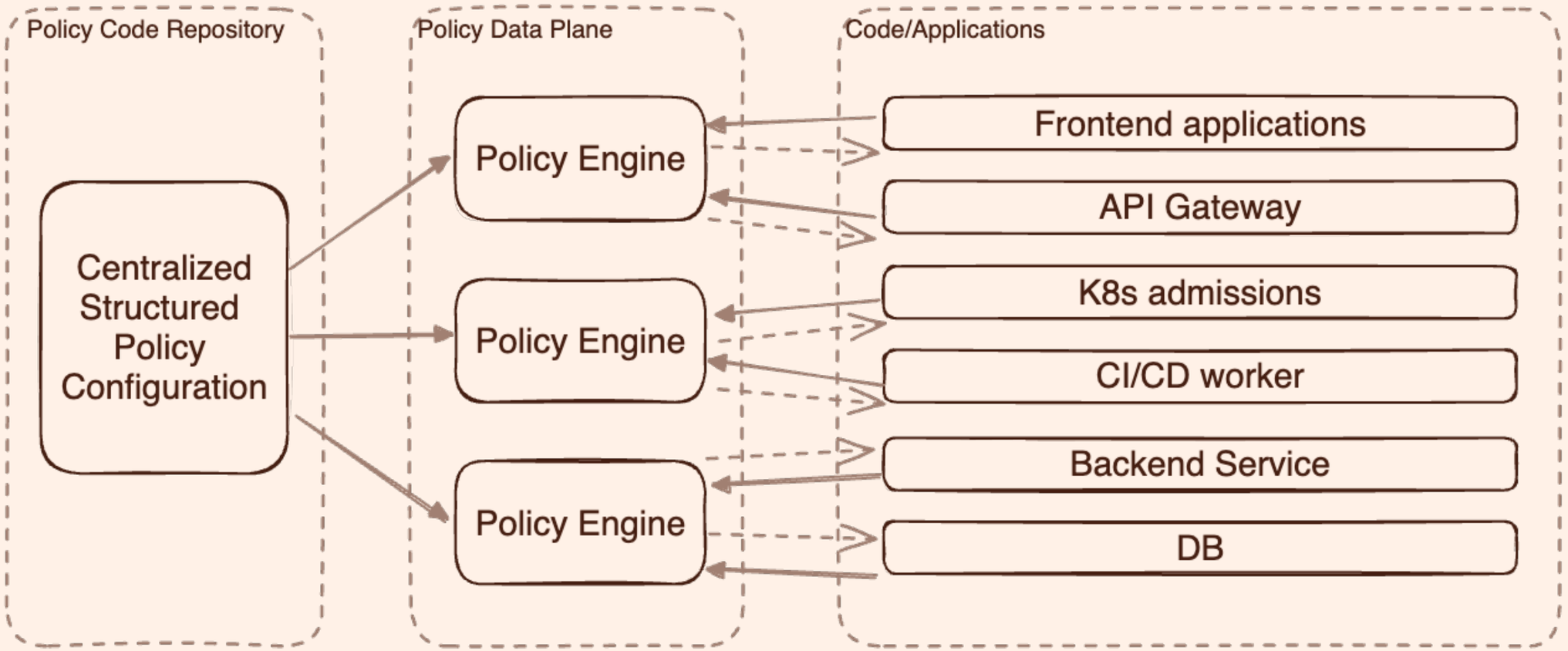
HCL

Chaos 👉 Structure

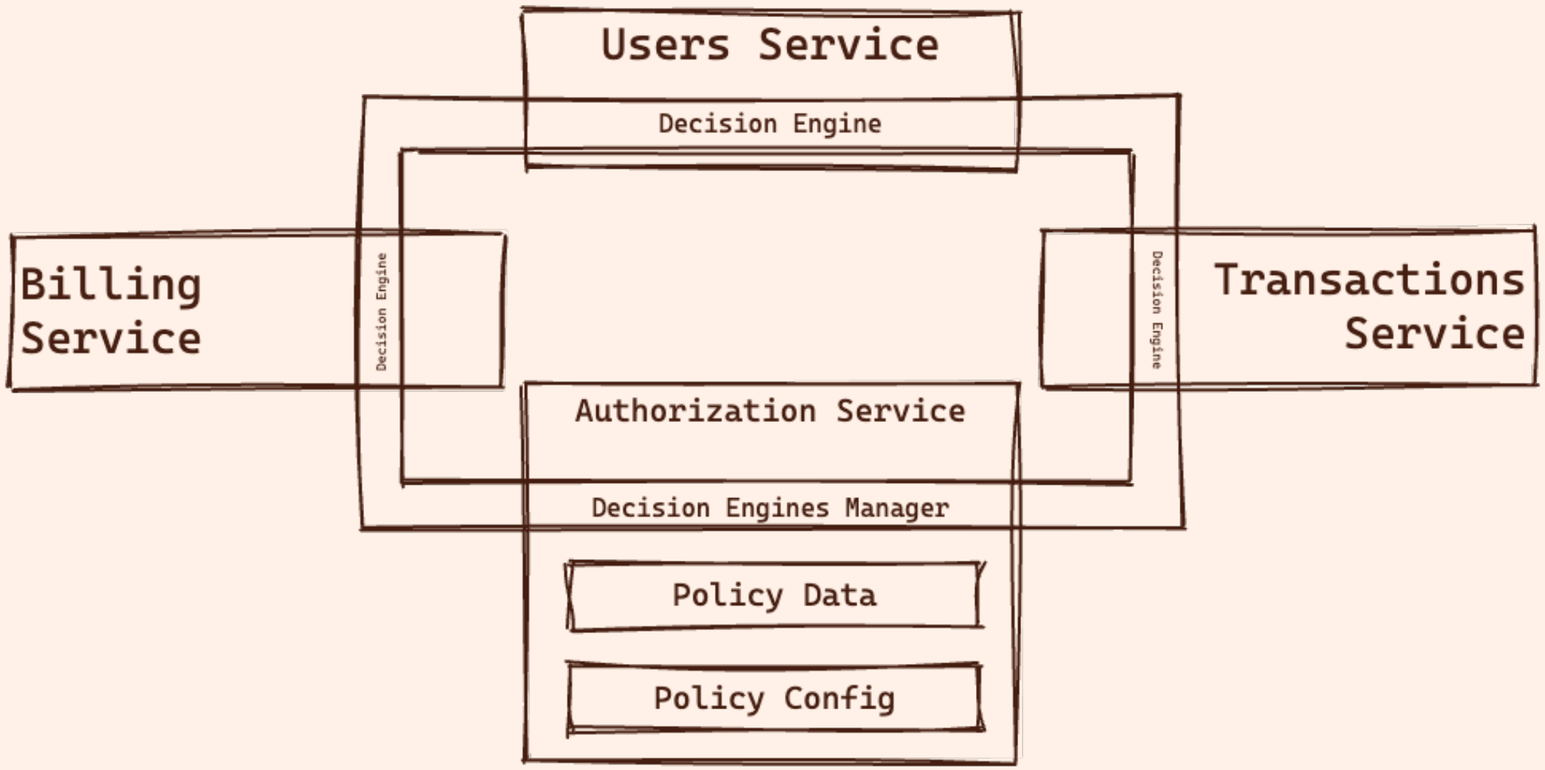
Authorization Layers



Abstract Authorization Service Model

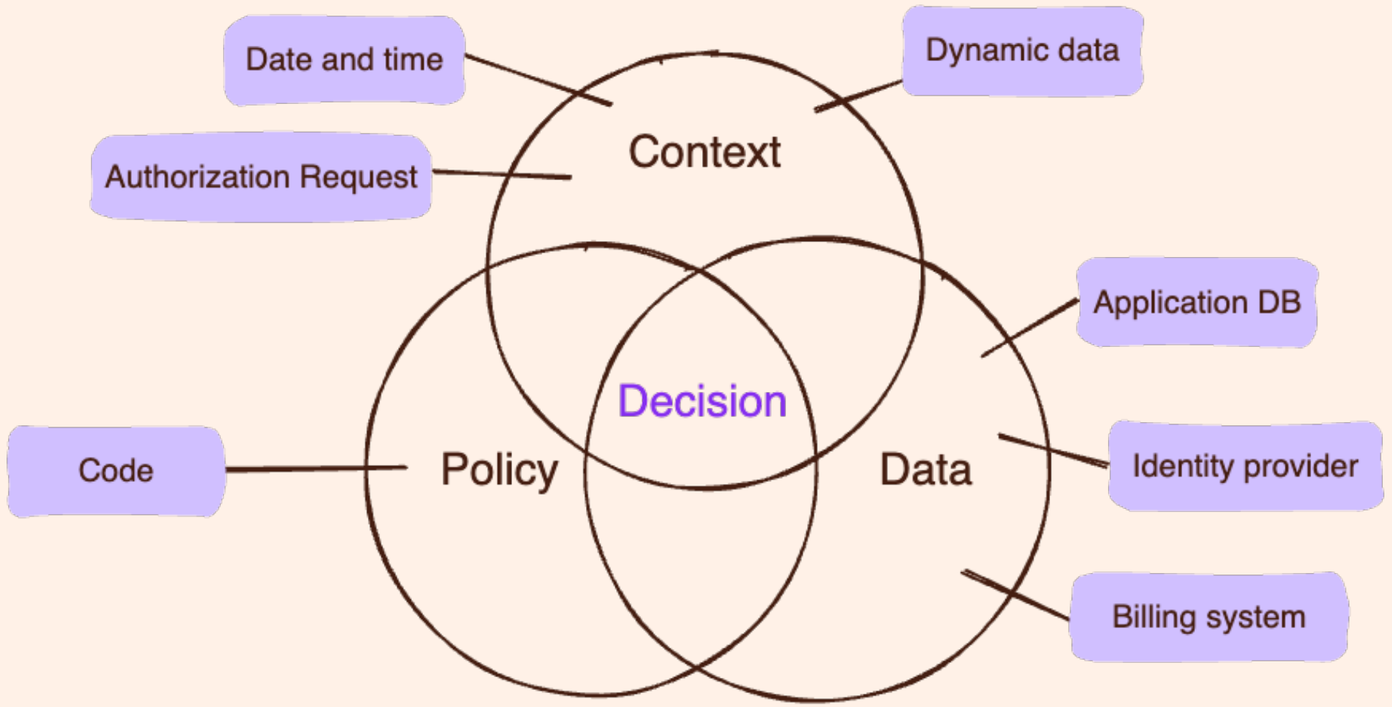


Application-Level Domain-Driven Model



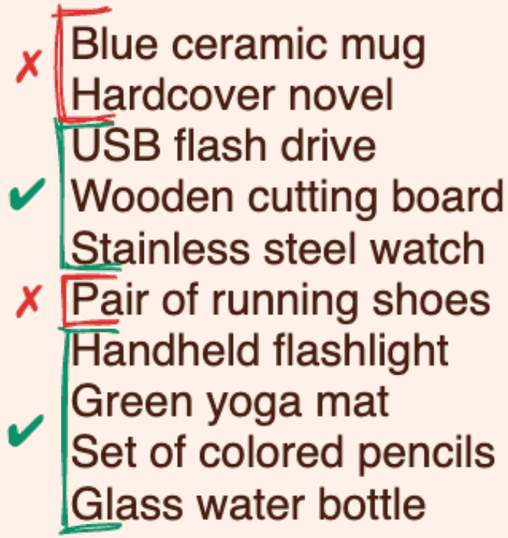
Complexity 👉 Design Patterns

Abstract Decision Components



Authorization Decision Types

Allow / Deny



Binary Check Function

```
check({ user }, { action }, { resource }, { context });
```

Binary Check Function

```
check({ user }, { action }, { resource }, { context });
```

```
response = requests.post('http://host.docker.internal:8180/v1/is_authorized', json={
    "principal": f"User::\"{user}\"",
    "action": f"Action::\"{method.lower()}\"",
    "resource": f"ResourceType::\"{original_url.split('/')[1]}\"",
    "context": request.json
})
```

```
const response = await fetch(
  "http://host.docker.internal:8180/v1/is_authorized",
  {
    method: "POST",
    body: JSON.stringify({
      principal: `User::\"${user}\"`,
      action: `Action::\"${method.toLowerCase()}\"`,
      resource: `ResourceType::\"${originalUrl.split("/")[1]}\"`,
      context: body,
    }),
  }
);
```

Partial Evaluation Flow

```
permit (
    principal == PhotoApp::User::"stacey",
    action == PhotoApp::Action::"viewPhoto",
    resource
)
when { resource in PhotoApp::Account::"stacey" };
```

⬇⬇⬇

```
evaluate("pictures", "staty", "viewPhoto");
```

⬇⬇⬇

```
SELECT *
FROM photos
WHERE owner = 'Stacey';
```

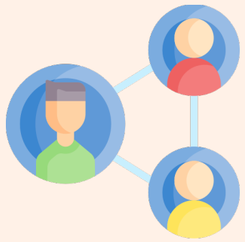
Velocity 👉 Frameworks

Permission Models

PBAC

RBAC

ABAC



ReBAC

Policy Based Access Control

✔ No External Data Policies

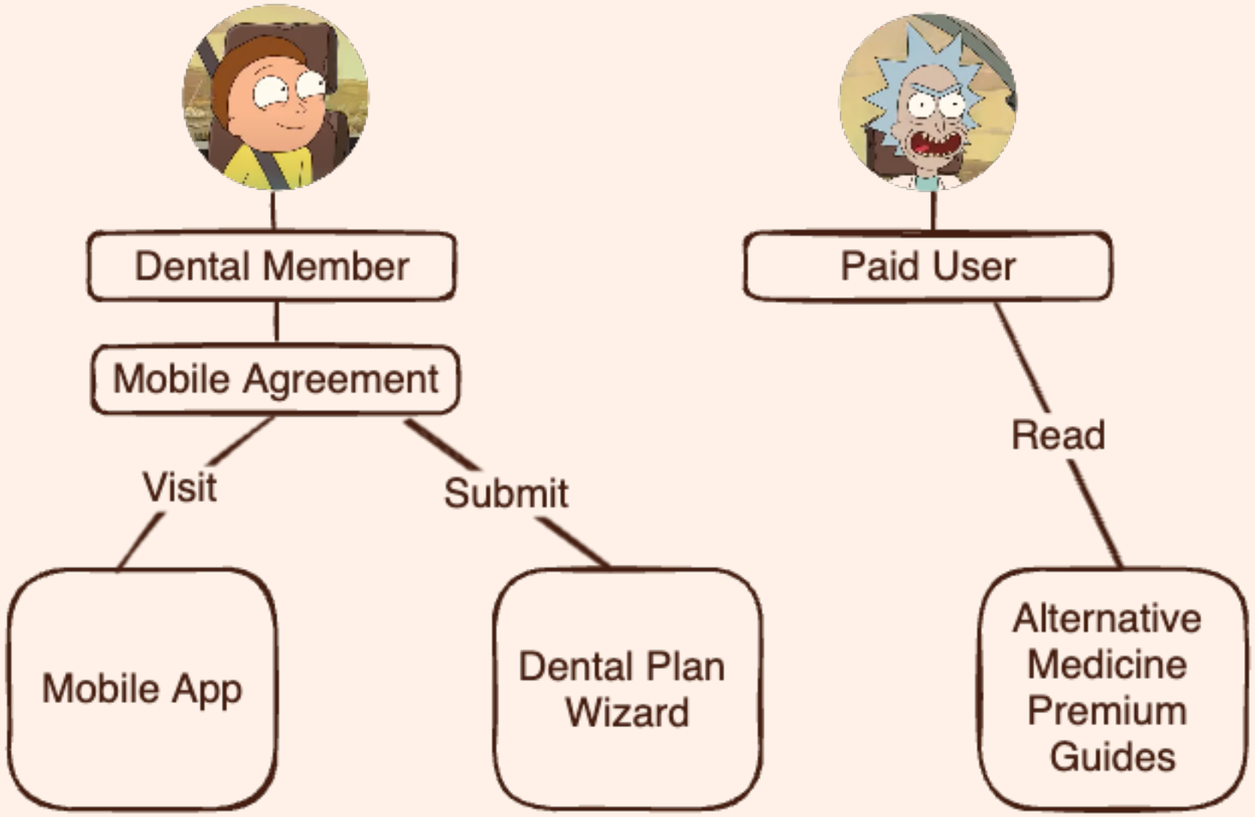
```
allow developers to deploy to production only on weekends

deny developers from creating S3 buckets with the name "prod"

allow developers to create S3 buckets only if their name starts with "dev-"
```

✗ External Data Policies

```
allow developers to deploy S3 buckets only if the next billing cycle is less than $100
```

Role Based Access Control

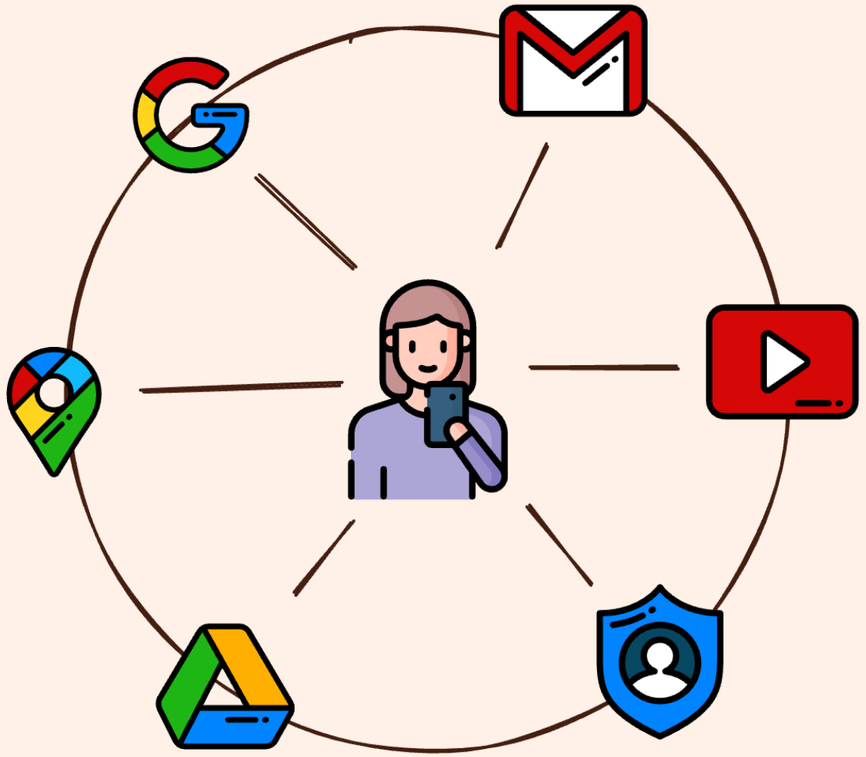
- Leverage user roles to determine permissions
- Commonly used as a term for authorization
- Very simple to implement and audit
- Flexible to use in different domains
- Hard to consider external data in policies
- Limited granularity in resource permissions


Dental Member
Mobile Agreement
Visit
Submit
Mobile App
Dental Plan Wizard
Paid User
Read
Alternative Medicine Premium Guides

Relationship Based Access Control

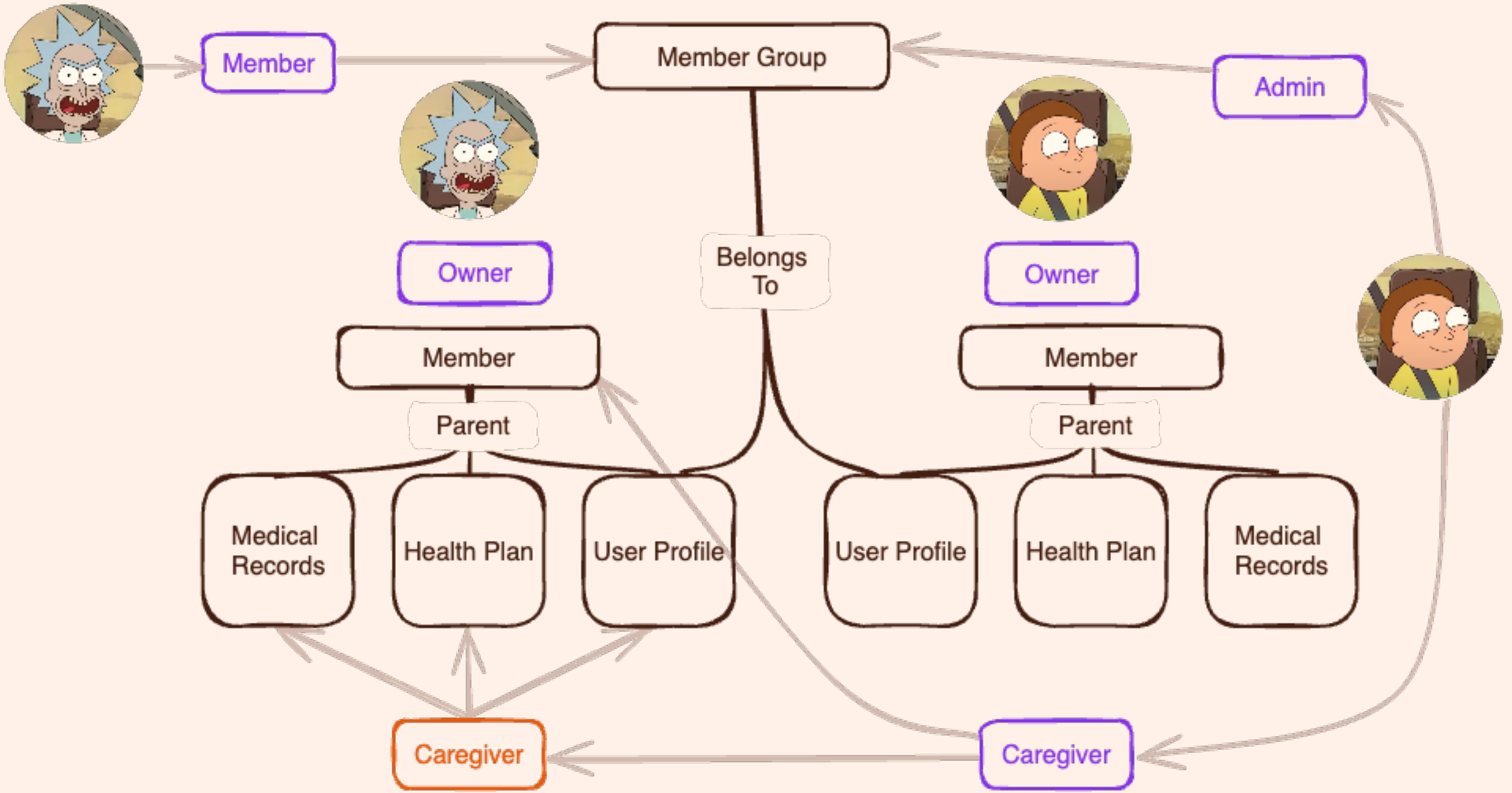
- Assign instance-level roles to users
- Leverage entity relationships to derive role permissions
- Built to support modern data models in efficient way
- Require centralized graph to calculate decisions
- Limited granularity in user and resource inspection

Google Multi Domain Authorization Challenge



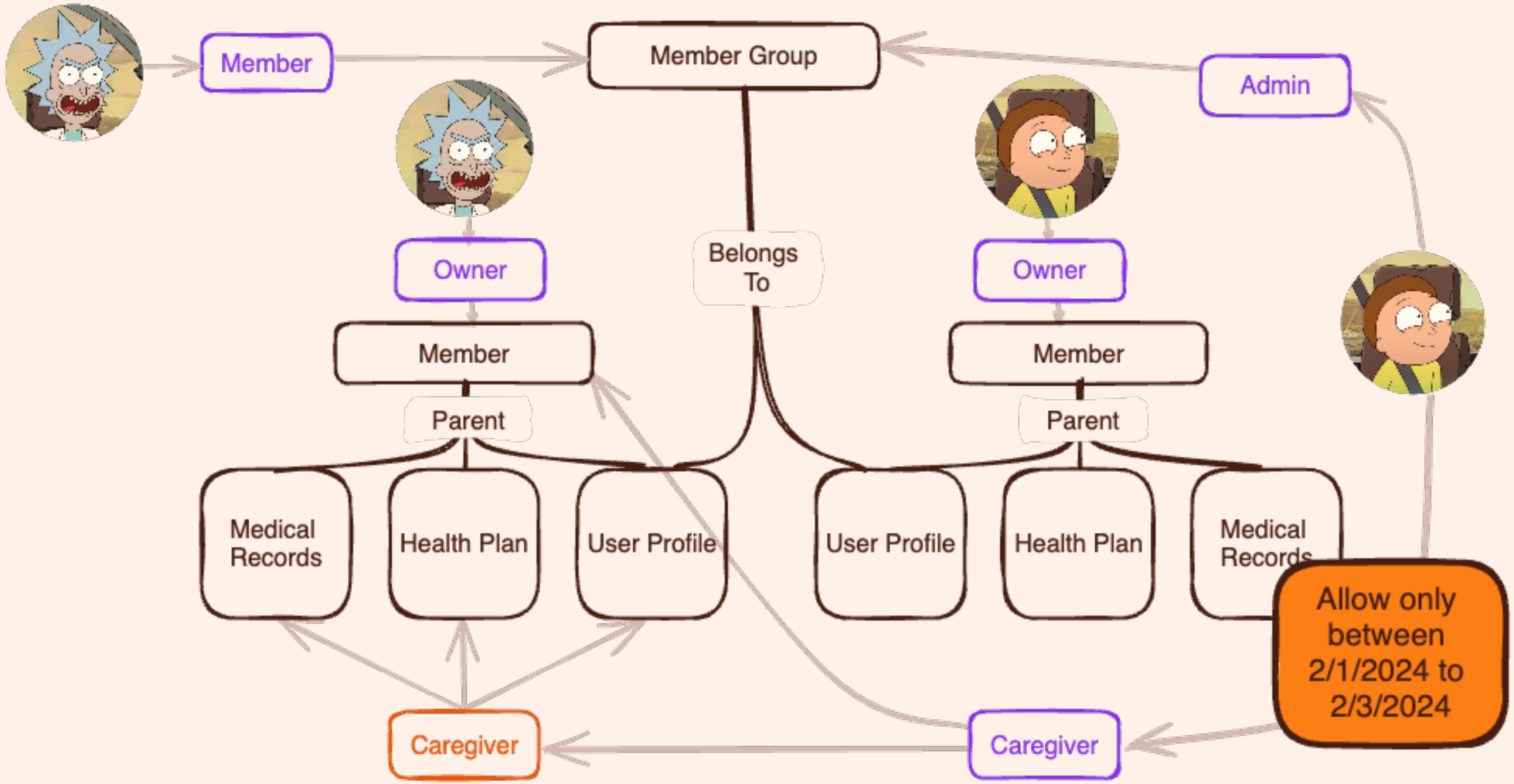
Google Zanzibar Principles

- Relationship Tuples
- Role Derivation
- Tuples Graph
- Relationship Based ACLs

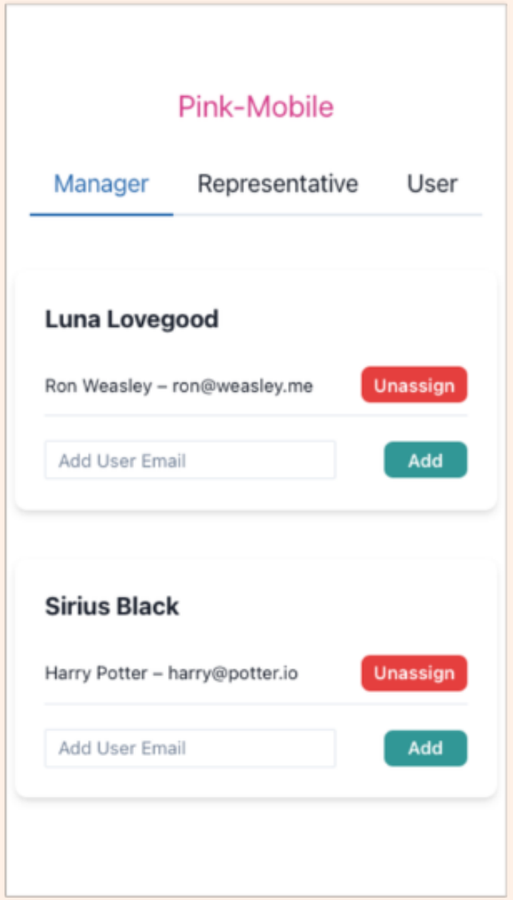

Member
Member Group
Admin
Owner
Belongs To
Owner
Member
Member
Parent
Parent
Medical Records
Health Plan
User Profile
User Profile
Health Plan
Medical Records
Caregiver
Caregiver

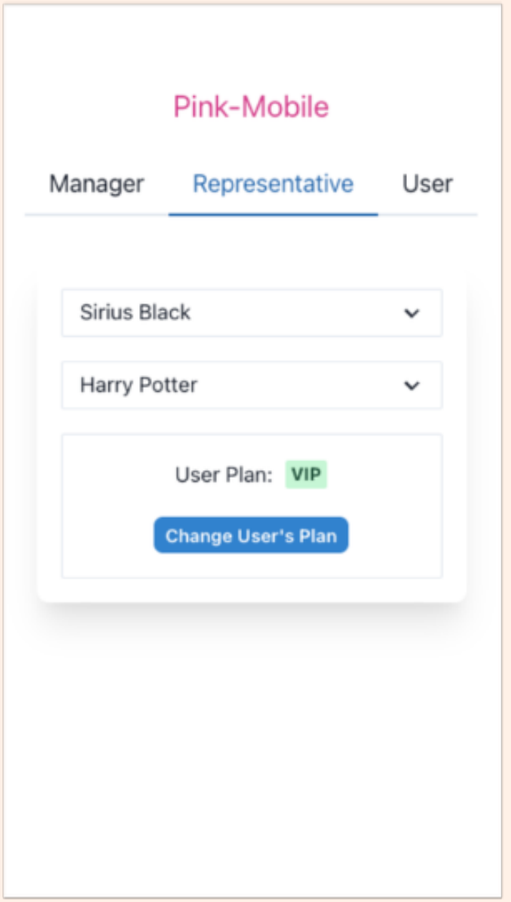
Attribute Based Access Control

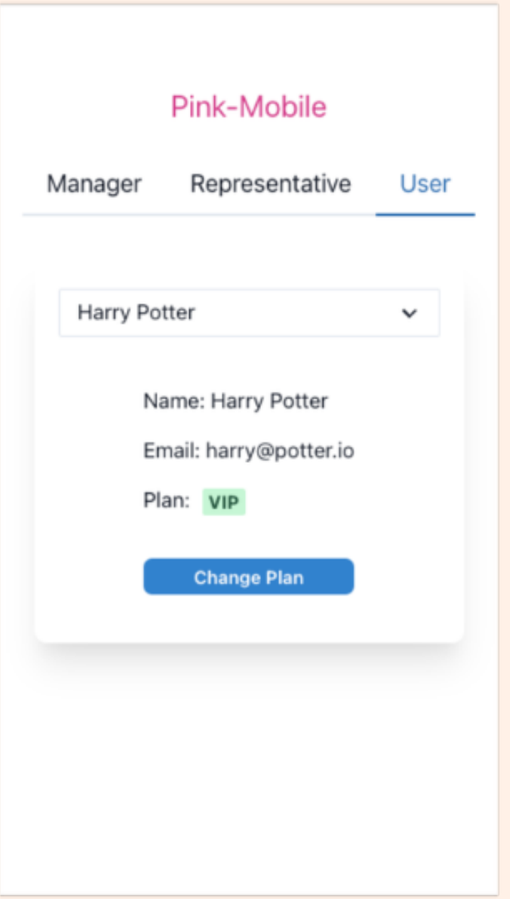
- Leverage entities' attributes to model permissions
- Use attribute conditions in the policies
- Most flexible structured policy model
- Complex to build and audit in imperative languages
- Usually used as extension to RBAC and ReBAC systems


Member
Member Group
Admin
Owner
Belongs To
Owner
Member
Member
Parent
Parent
Medical Records
Health Plan
User Profile
User Profile
Health Plan
Medical Records
Allow only between 2/1/2024 to 2/3/2024
Caregiver
Caregiver

io.permit.io/pink-mobile

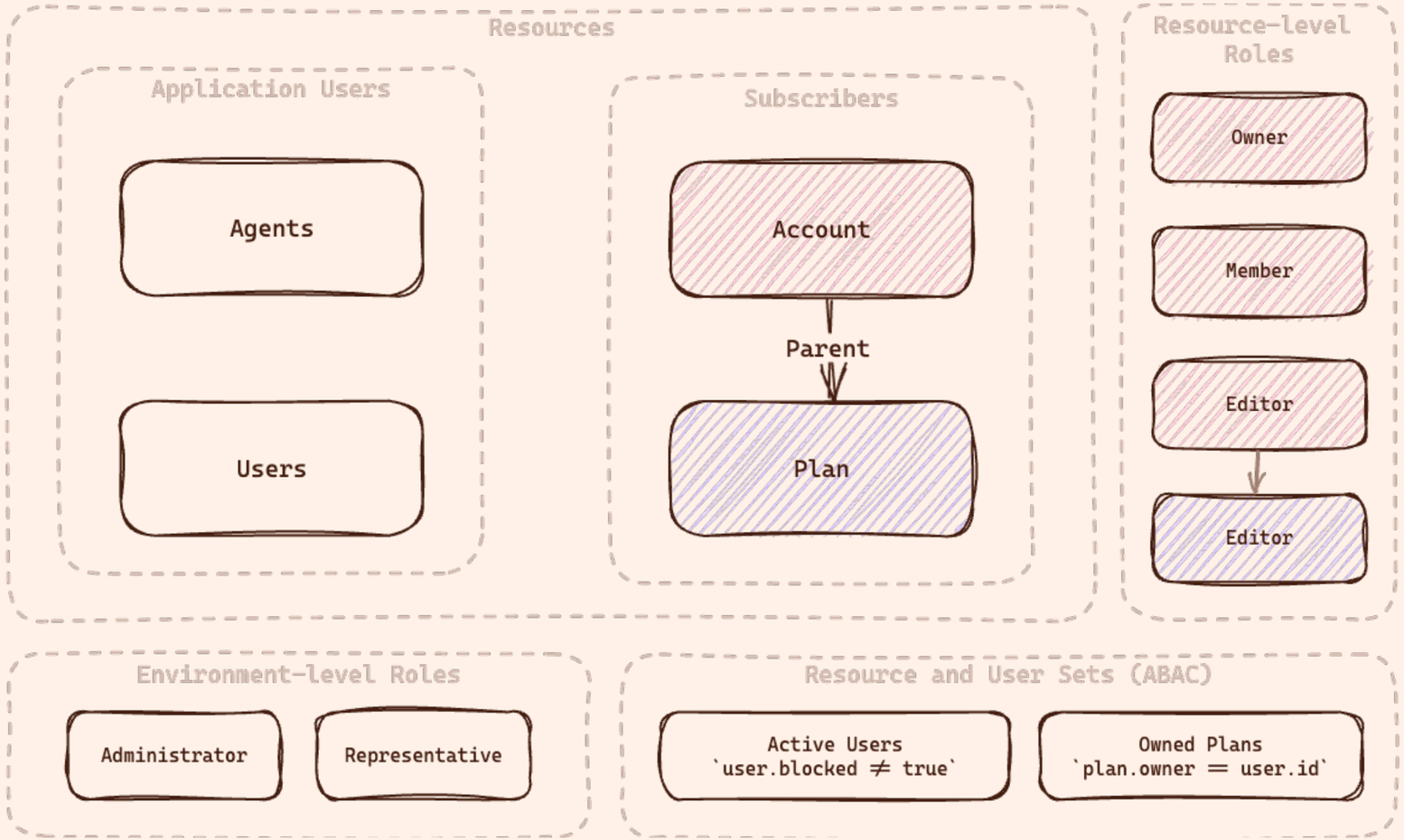






Permit.io

@gemanor



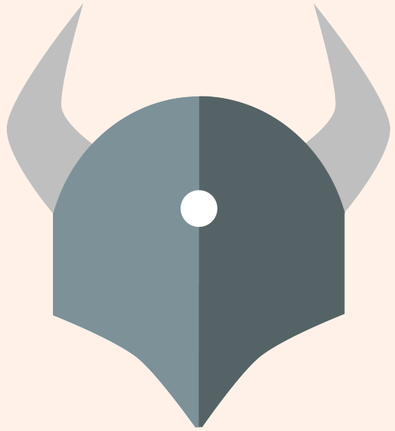
RBAC, ReBAC, ABAC Modeling Example

Confidence 👉 Domain-Specific Declarative Code

```
// Users can edit their own info, admins can edit anyone's info
permit (
    principal,
    action,
    resource in HealthCareApp::InfoType::"accountinfo"
)
when {
    resource.subject == principal ||
    principal in HealthCareApp::Role::"admin"
};

//A patient may create an appointment for themselves, or an administrator can do it
permit (
    principal,
    action == HealthCareApp::Action::"createAppointment",
    resource
)
when {
    (context.referrer in HealthCareApp::Role::"doctor"  && resource.patient == principal) ||
    principal in HealthCareApp::Role::"admin"
};
```

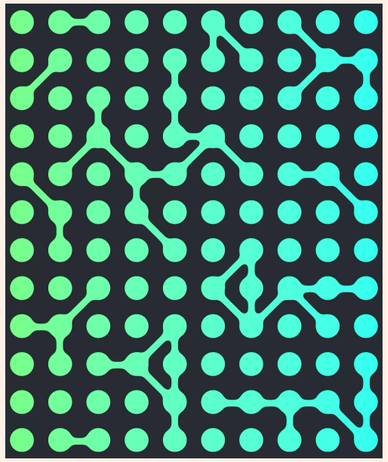
Domain-Specific Declarative Code Advantages

- Performance
- Reusability
- Versioning
- Testing
- Maintainance
- Readability
- Validation



Open Policy Agent

AWS Cedar



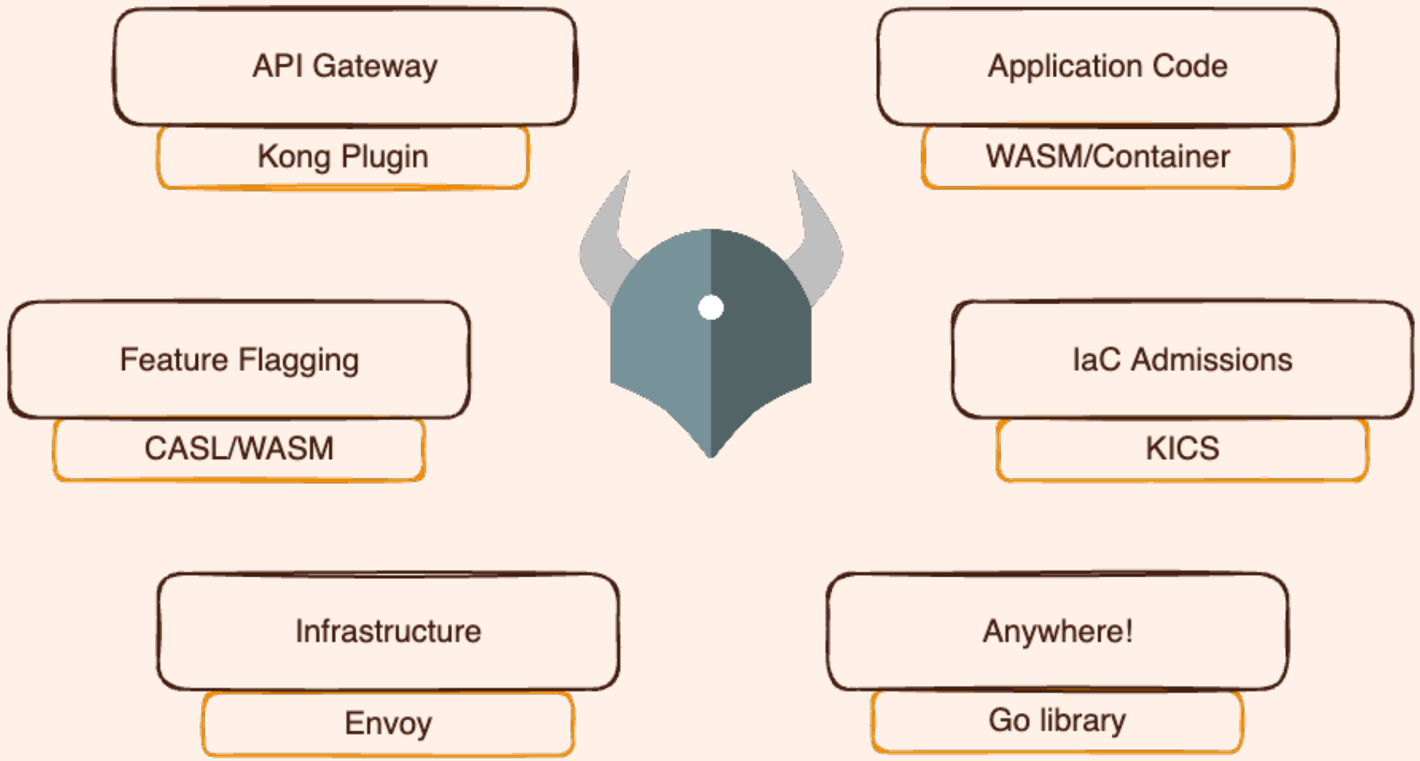
OpenFGA

```
allow {
    input.user.role == "viewer"
    validate_department(input.user, input.document)
    validate_classification(input.user.role, input.document.classification)
    validate_dynamic_rules(input.user, input.document)
}

validate_department(user, document) {
    user.department == document.department
}

validate_classification(user_role, doc_classification) {
    role_permissions[user_role][_] == doc_classification
}

validate_dynamic_rules(user, document) {
    dynamic_rules[_](user, document)
}
```

Open Policy Agent

- Started as a multi-purpose policy engine
- Using Rego as a policy language
- As part of their GTM, they are focusing on Kubernetes admission
- Works great for application-level authorization (<10 ms for decision)
- Hard to deal with large (>2gb) sets of data
- Need some tweak for efficient ReBAC support

OPA Multi Domain Support



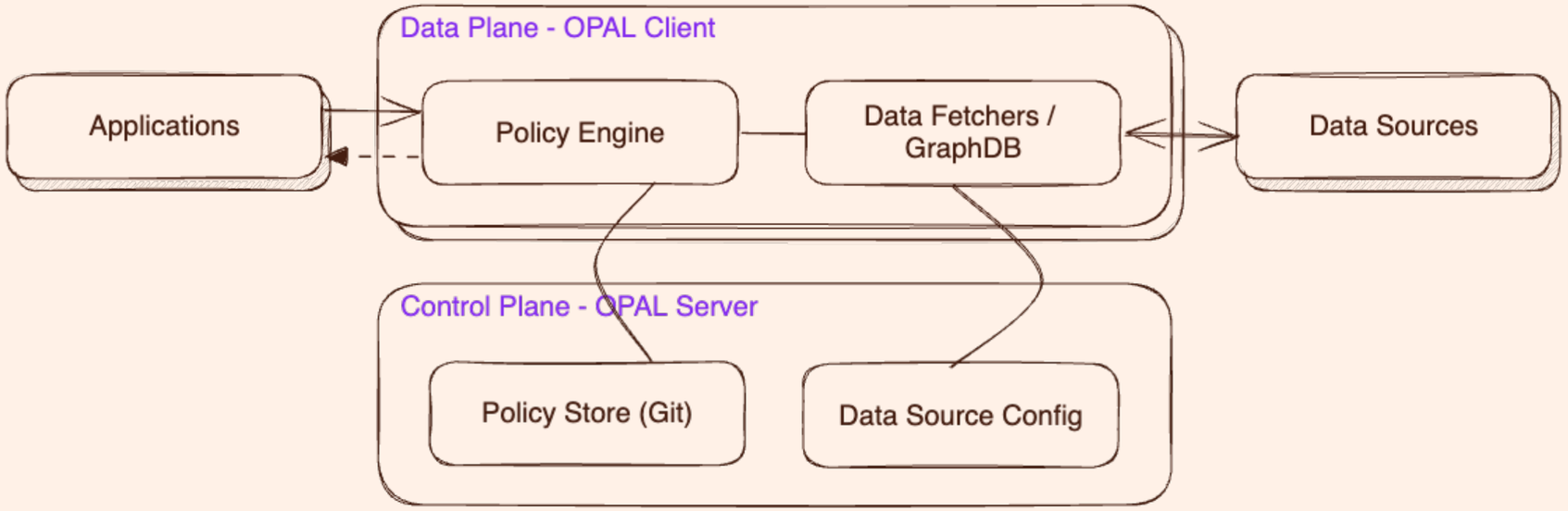
```
permit (
    principal == PhotoApp::User::"stacey",
    action == PhotoApp::Action::"viewPhoto",
    resource
)
when { resource in PhotoApp::Account::"stacey" };
```

AWS Cedar

- Started as a language for application-level authorization
- Not like AWS IAM, it's a language that can be used in any application
- AWS released it as open source, but also offer it as a service (AVP)
- Use Dafny language to provide scientific proof for correctness and performance
- (Still) hard to deal with unstructured data - no real ReBAC support
- Faster option for ABAC decisions
- Support audit, static analysis, and partial evaluation out of box

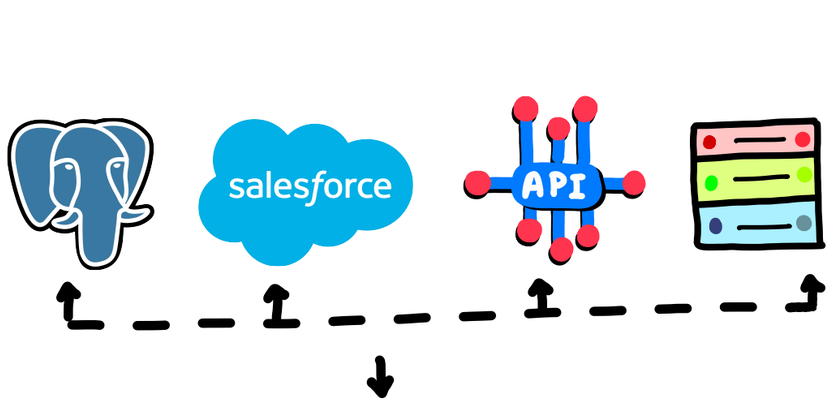
Company	Language	Models
Netflix	Rego	RBAC, ABAC, ReBAC
AirBnB	Himeji (Zanzibar)	ReBAC
Uber	CEL	ABAC
Google	Zanzibar	ReBAC
Reddit	Rego	RBAC, ABAC
AWS	Cedar	RBAC, ABAC

OPAL - Open Policy Administration Layer



An End-to-End *Batteries Included* Authorization Service

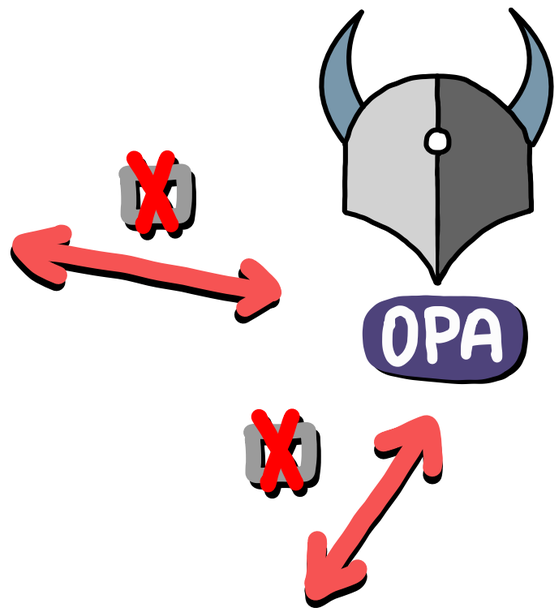
Gaps with OPA ??



Data Sources

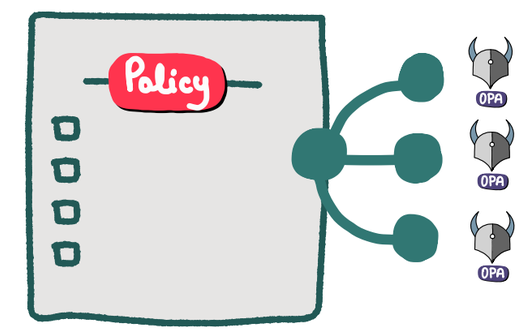
Integration

If data needed for authorization say user's list is needed to be pulled from DB then that is not possible with OPA.



Policy Sync

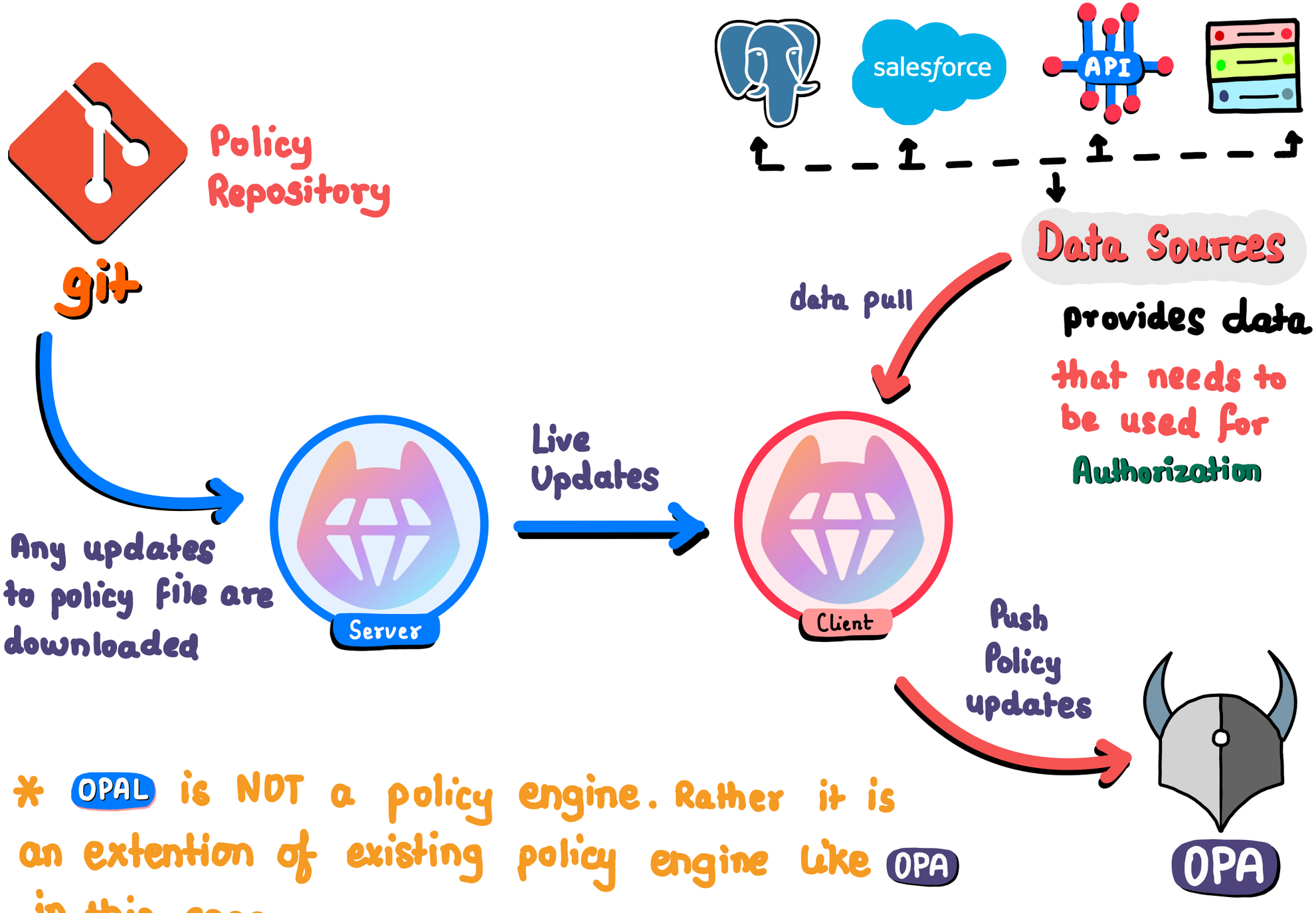
Real time policy Sync is not possible with OPA !!



Management

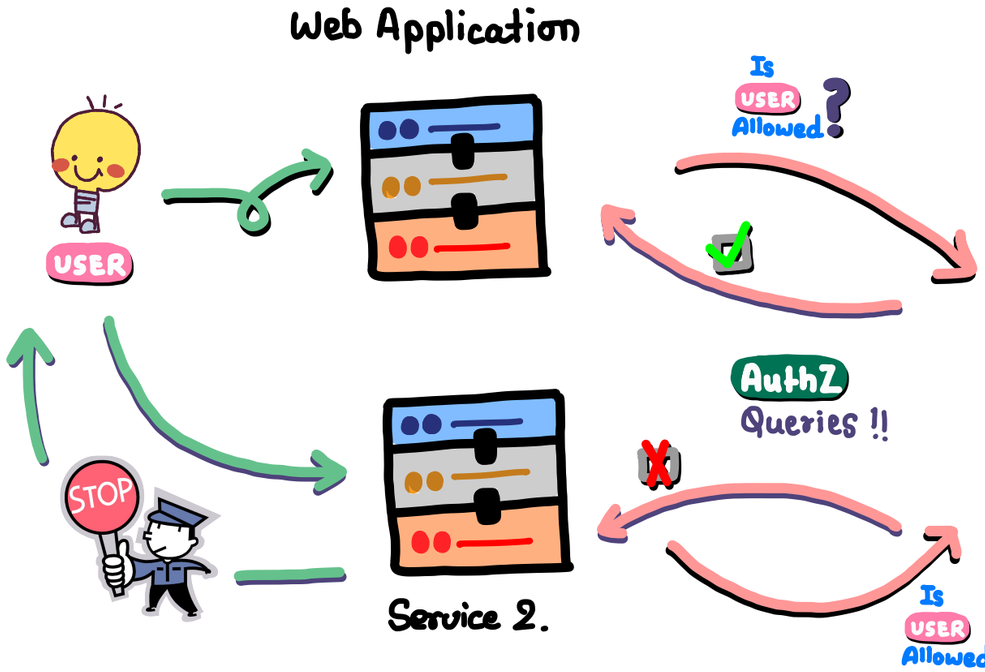
If Same policy has to be shared with multiple agents then this has to be done manually.

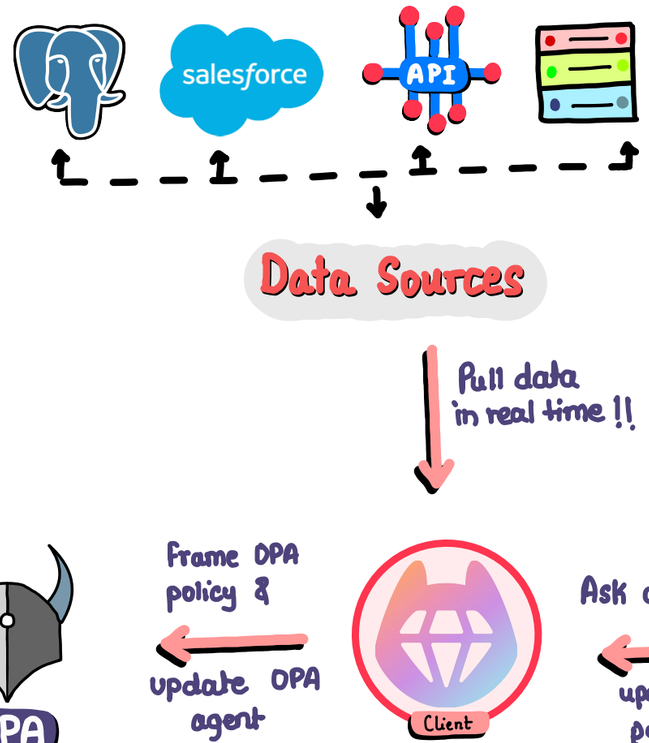
OPAL A Client - Server Model

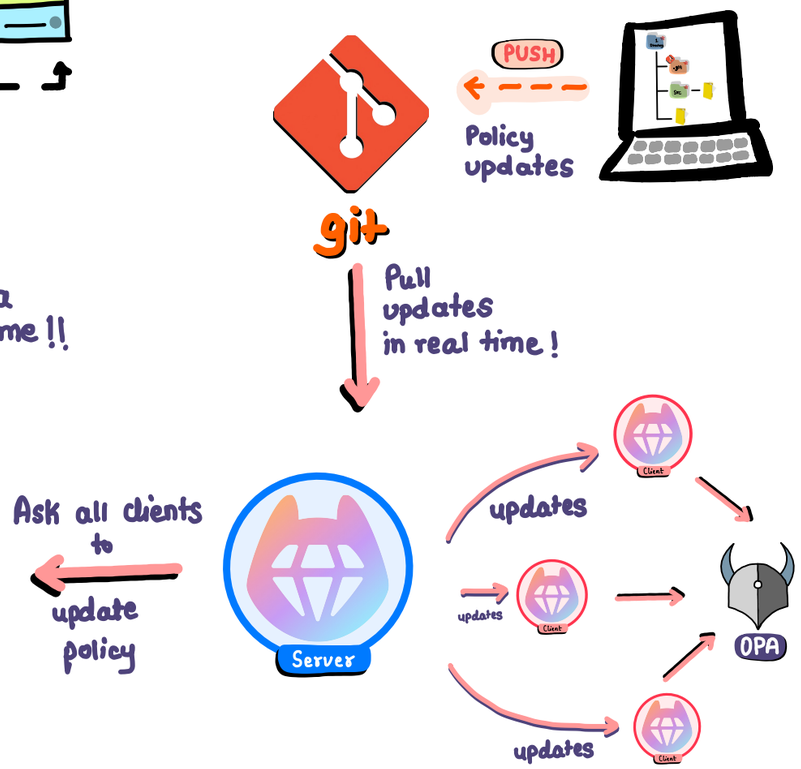


* OPAL is NOT a policy engine. Rather it is an extention of existing policy engine like OPA in this case.

OPAL Workflow !!







OPAL Specifications

- End-to-End Authorization
- Policy Engine Plugins
- GitOps Policy Flow
- Data Sharding ReBAC
- Centralized Config
- Decentralized Decision
- Pub/Sub Policy & Data Updates
- Extensible Data Fetchers
- Plugable Data Sources

Thank You 🙏

Join OPAL Slack Community *io.permit.io/opal-slack*