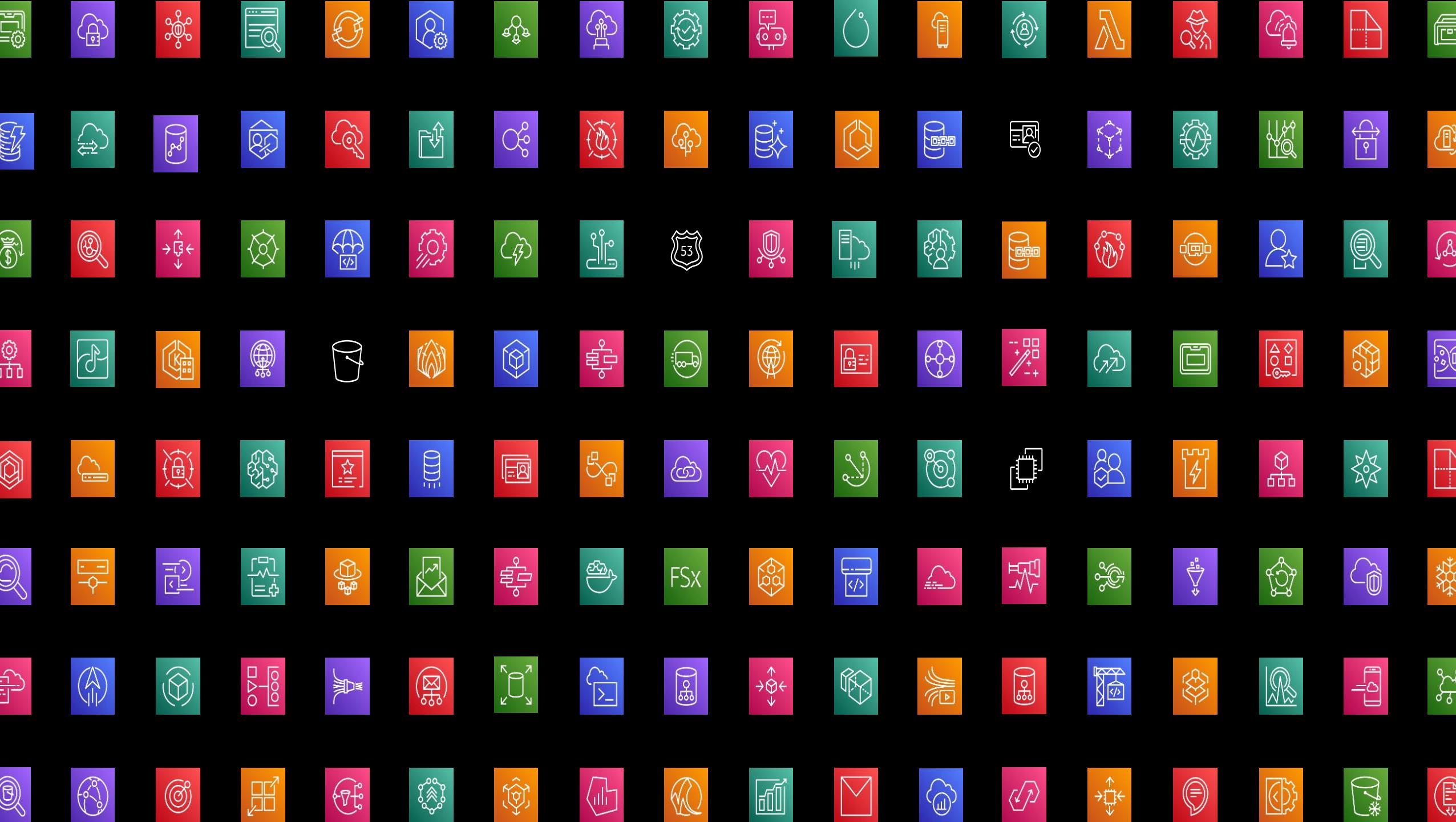


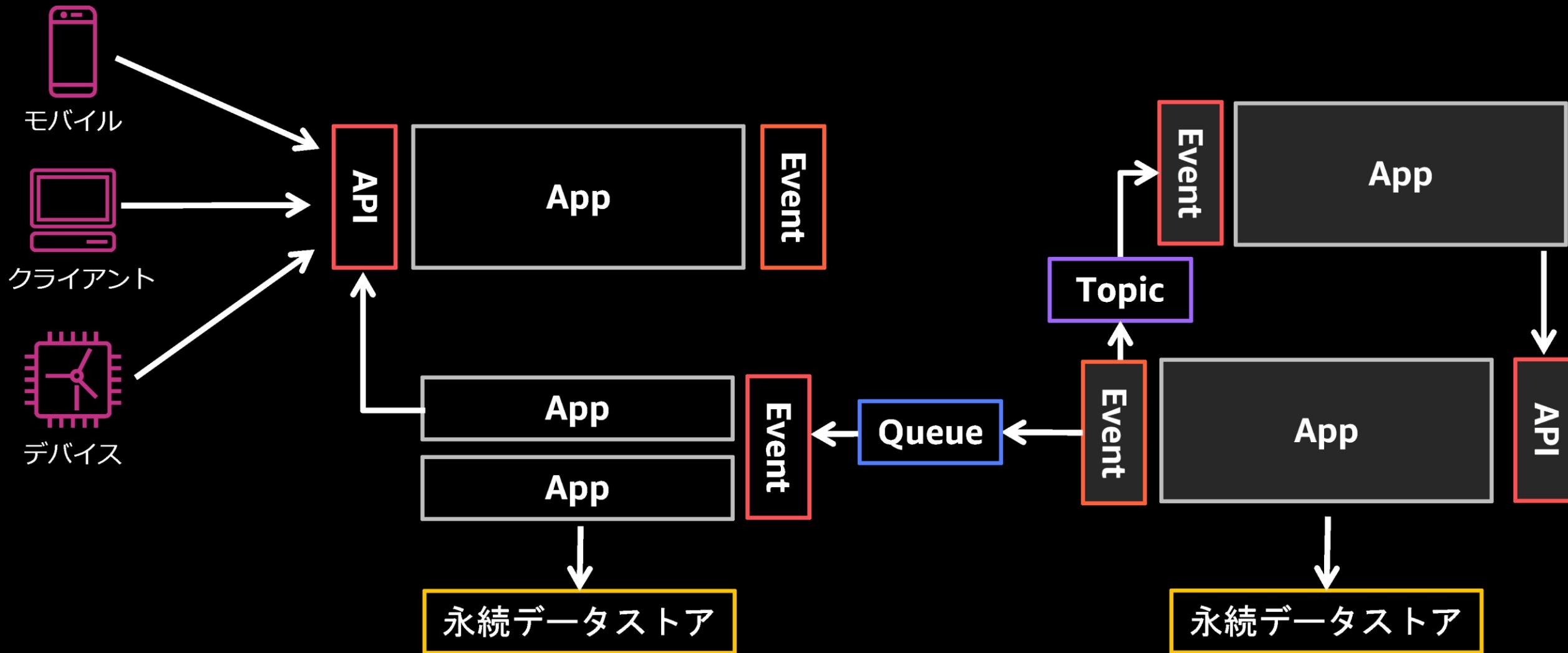
CloudflareのClickHouse活用事例



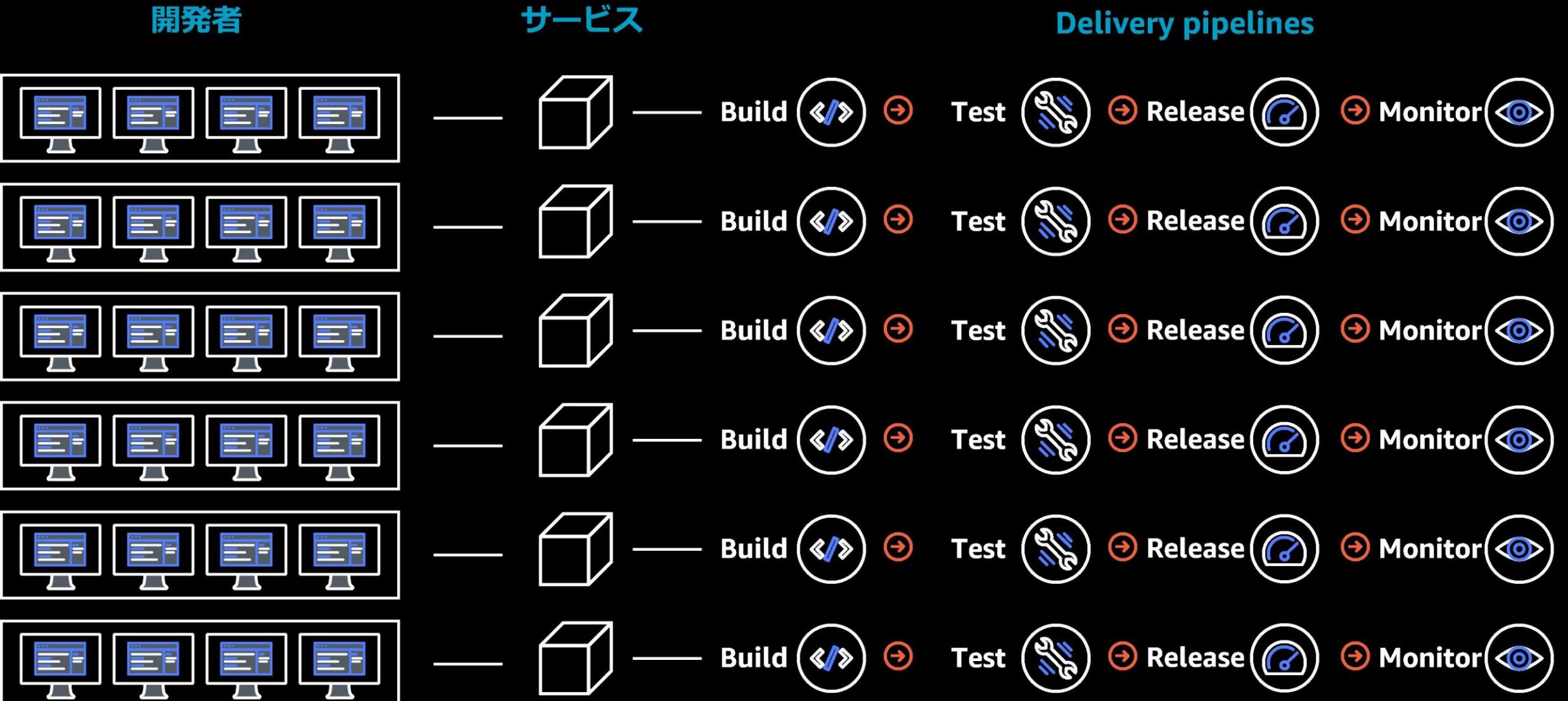
Harunobu Kameda
Evangelist @ Cloudflare Japan



EDA (Event Driven Architecture)



Microservice Architecture





ガードレール

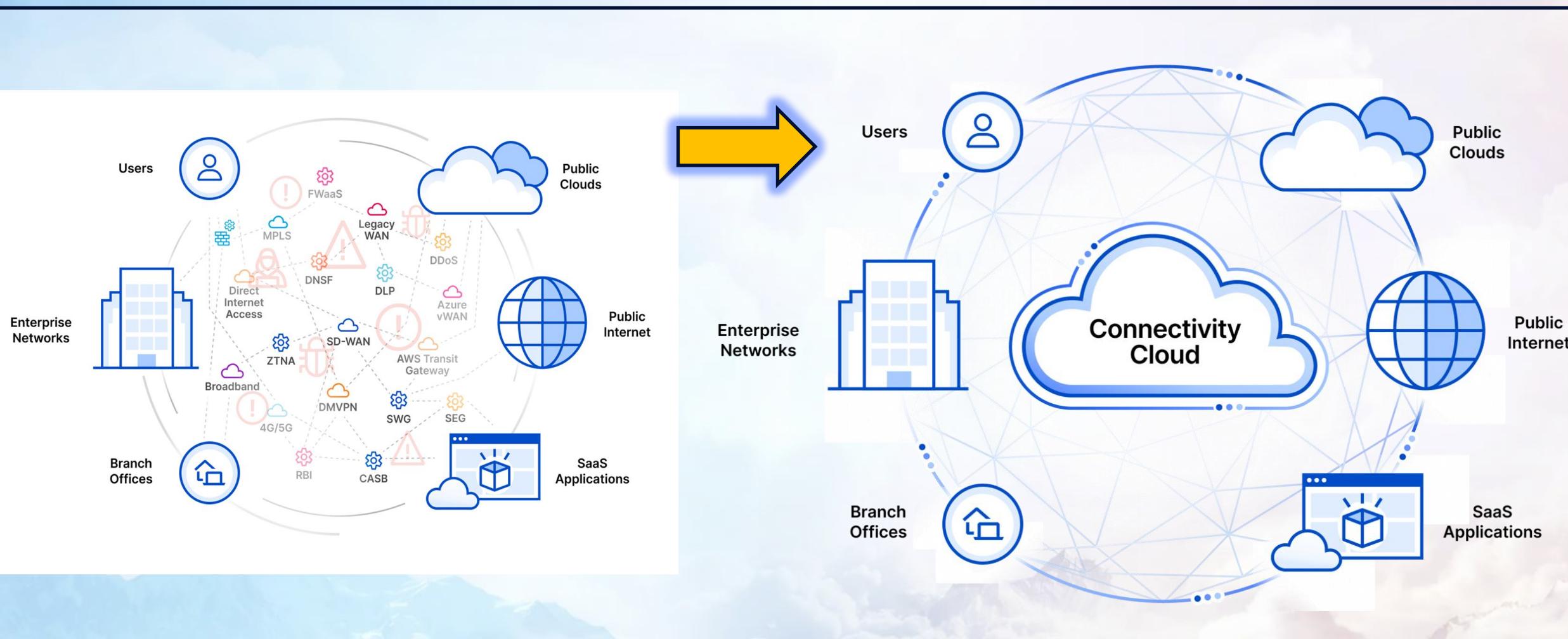
(Guard Rail)

Shift Left の実現

SHIFTLIMIT.COM

分散と集中を繰り返す IT

『攻撃表面』 『ミスの混入』 個所を集中化させることで、自由な分散を実現



Network は将来必ず企業の成長阻害要因になる

Infrastructure complexity

High maintenance cost

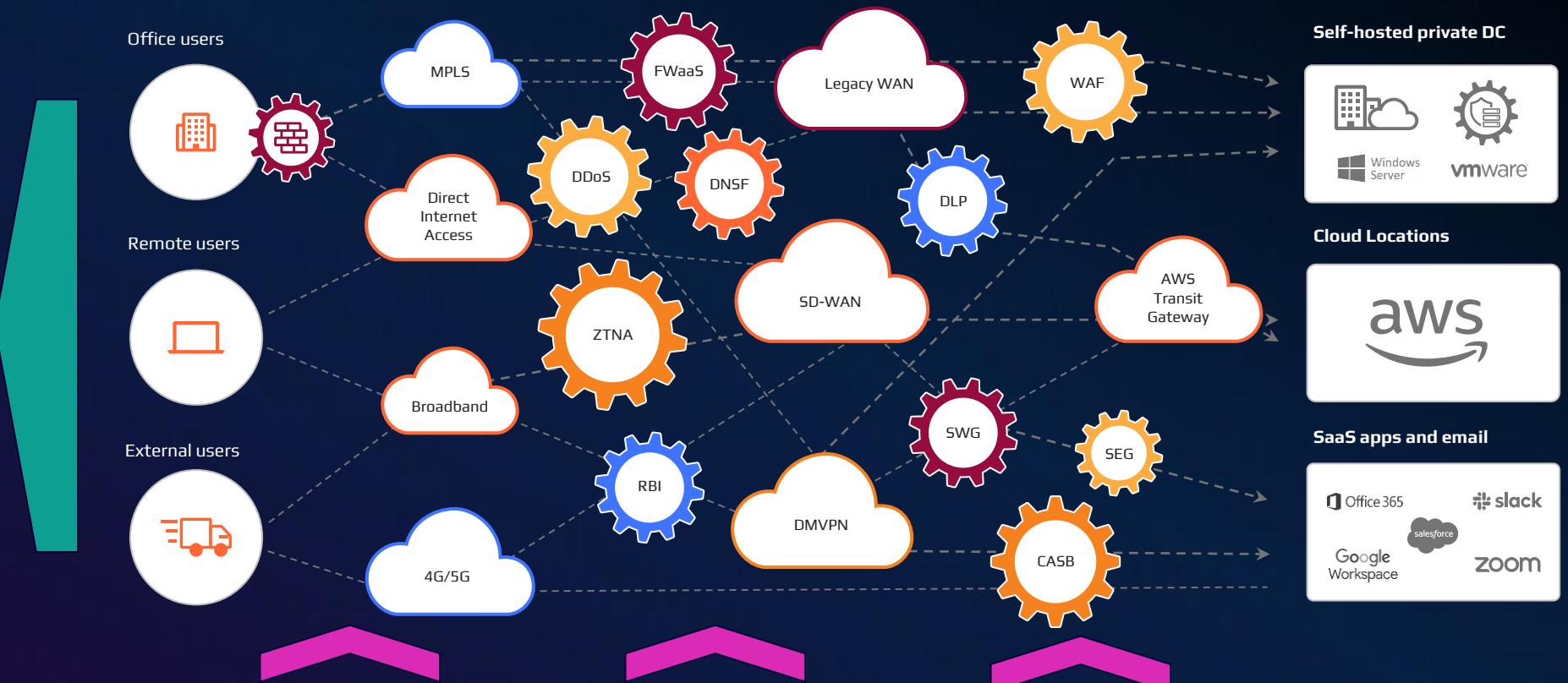
Performance challenges

下がるメンテナンス性とパフォーマンス+増大するセキュリティ予算

ラストマイル

個別最適の
テクノロジー

ファーストマイル



Supply Chain Attack

前年順位	個人	順位	組織	前年順位
1位	フィッシングによる個人情報等の詐取	1位	ランサムウェアによる被害	1位
2位	ネット上の誹謗・中傷・デマ	2位	サプライチェーンの弱点を悪用した攻撃	3位
3位	メールやSMS等を使った脅迫・詐欺の手口による金銭要求	3位	標的型攻撃による機密情報の窃取	2位
4位	クレジットカード情報の不正利用	4位	内部不正による情報漏えい	5位
5位	スマホ決済の不正利用	5位	テレワーク等のニューノーマルな働き方を狙った攻撃	4位
7位	不正アプリによるスマートフォン利用者への被害	6位	修正プログラムの公開前を狙う攻撃(ゼロデイ攻撃)	7位
6位	偽警告によるインターネット詐欺	7位	ビジネスメール詐欺による金銭被害	8位
8位	インターネット上のサービスからの個人情報の窃取	8位	脆弱性対策の公開に伴う悪用増加	6位
10位	インターネット上のサービスへの不正ログイン	9位	不注意による情報漏えい等の被害	10位
国外	ワンクリック請求等の不正請求による金銭被害	10位	犯罪のビジネス化(アンダーグラウンドサービス)	国外

企業の情報漏洩事案の
91%は電子メール 経由

サプライチェーンを狙った
攻撃も新しい侵入経路に
(インターネットを介した
システム連携が攻撃対象に)

<https://www.ipa.go.jp/security/10threats/10threats2023.html>



Connectivity Cloud



Cloudflare's global network



320 Cities

120 カ国

13,000 Networks

主要ISP、クラウドサービスへの相互直接接続

280 Tbps

トランジット接続、ピアリング、
プライベートネットワーク相互接続で構成される
グローバルネットワークのエッジ容量

~50 ms

世界のインターネット接続人口の95%

CDN / WAF のログの特徴

1. 数千万/秒 のリクエスト

2. カスタムヘッダによる不定スキーマ

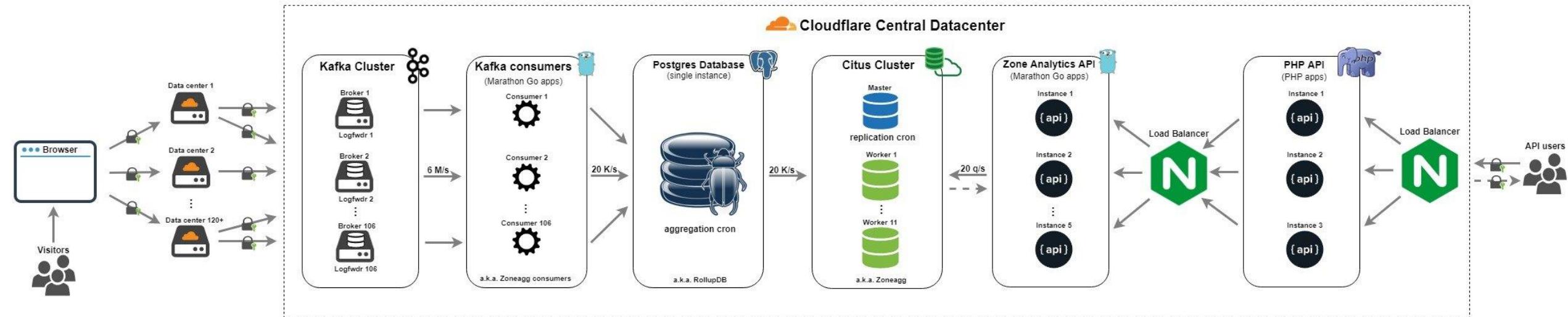
3. 分散アーキテクチャとデータパイプライン

CDN / WAF のログの特徴

世界中のノードから集約されるログ

Legacy なアーキテクチャ (in 2014)

Old Pipeline Architecture



SPOF とスケーラビリティ

CDN / WAF のログの特徴

1回の通信で処理されるログ

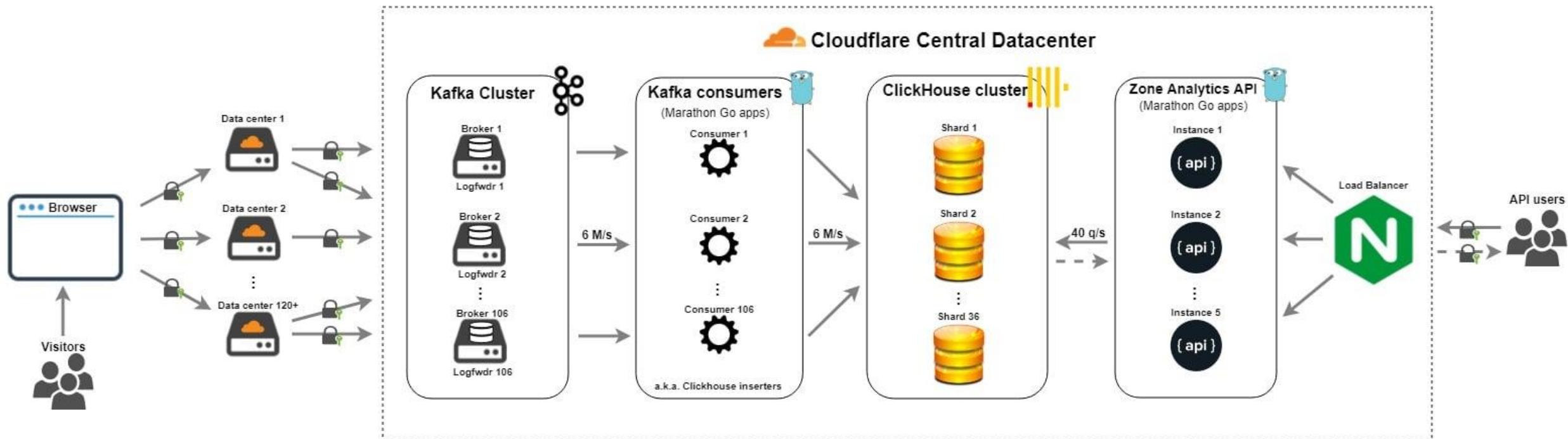
```
{  
    "CacheCacheStatus": "dynamic",  
    "CacheResponseBytes": 8793,  
    "CacheTieredFill": false,  
    "ClientASN": 2527,  
    "ClientCountry": "jp",  
    "ClientDeviceType": "desktop",  
    "ClientIP": "240d:f:438:4300:70b0:cbb7:13d9:ea9b",  
    "ClientMTLSAuthStatus": "unknown",  
    "ClientRequestBytes": 3348,  
    "ClientRequestHost": "20240425-test2.a.harunobukameda.com",  
    "ClientRequestMethod": "GET",  
    "ClientRequestPath": "/",  
    "ClientRequestProtocol": "HTTP/2",  
    "ClientRequestReferer": "",  
    "ClientRequestScheme": "https",  
    "ClientRequestSource": "eyeball",  
    "ClientRequestURI": "/",  
    "ClientRequestUserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.0.0 Safari/537.36",  
    "ClientSSLCipher": "AEAD-AES128-GCM-SHA256",  
    "ClientSSLProtocol": "TLSv1.3",  
    "ClientSrcPort": 63472,  
    "ClientTCPRTTMs": 13,  
    "ClientXRequestedWith": "",  
    "Cookies": {}  
}
```

```
"EdgeCFConnectingO20": false,  
"EdgeColoCode": "NRT",  
"EdgeEndTimestamp": "2024-06-04T01:22:47Z",  
"EdgeResponseBodyBytes": 2807,  
"EdgeResponseBytes": 3150,  
"EdgeResponseContentType": "text/html",  
"EdgeResponseStatus": 200,  
"EdgeServerIP": "172.70.222.68",  
"EdgeStartTimestamp": "2024-06-04T01:22:47Z",  
"EdgeTimeToFirstByteMs": 69,  
"OriginDNSResponseTimeMs": 0,  
"SecurityRuleID": "",  
"OriginIP": "157.65.27.7",  
"OriginRequestHeaderSendDurationMs": 0,  
"OriginResponseDurationMs": 62,  
"OriginResponseHTTPExpires": "",  
"OriginResponseHTTPLastModified": "",  
"OriginResponseHeaderReceiveDurationMs": 60,  
"OriginResponseStatus": 200,  
"OriginSSLProtocol": "none",  
"OriginTCPHandshakeDurationMs": 2,  
"OriginTLSHandshakeDurationMs": 0,  
"ParentRayID": "00",  
"RayID": "88e41d25bcab80c3",  
"RequestHeaders": {},  
"ResponseHeaders": {},  
"SecurityAction": "",  
"SecurityActions": [],  
"SecurityRuleIDs": [],  
"SecuritySources": [],  
"SecurityRuleDescription": "",  
"SecurityLevel": "med",  
"SmartRouteColoID": 0,  
"UpperTierColoID": 0,  
"WorkerStatus": "unknown",  
"WorkerSubrequest": false,  
"WorkerSubrequestCount": 0,  
"sampleInterval": 1  
}
```

CDN / WAF のログの特徴

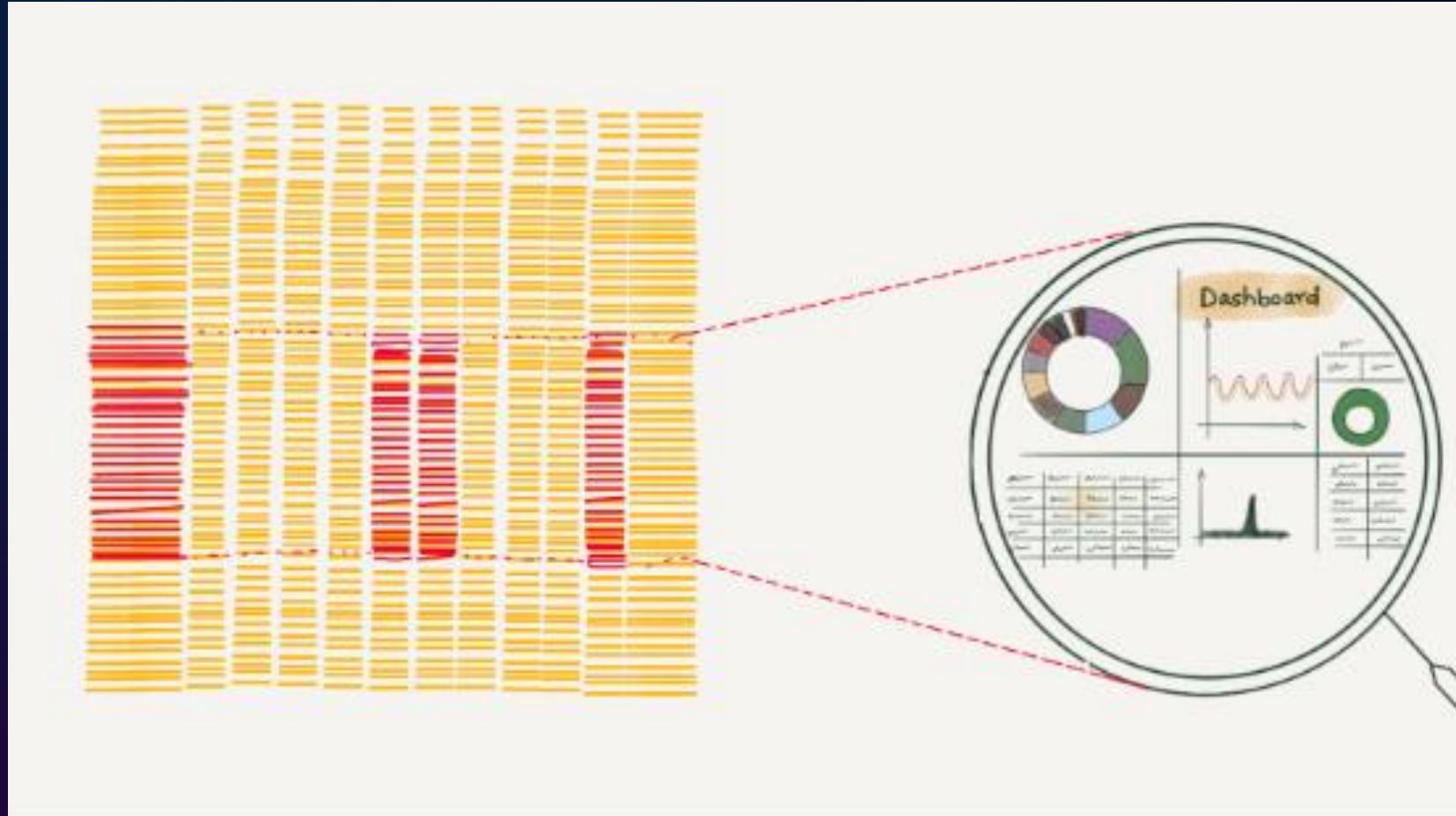
SPOF の排除と水平スケーリング

New Pipeline Architecture



CDN / WAF のログの特徴

殆どのログは似ている - 個別より集計へのフォーカス



Custom HTTP Header

アプリケーションで利用可能な
カスタムなHTTPヘッダ

ユーザーの制御やWAFで利用される

```
// Enter Snippet code below

export default {
  async fetch(request) {
    const response = await fetch(request);

    // Clone the response so that it's no longer immutable
    const newResponse = new Response(response.body,
response);

    // Add a custom header with a value
    newResponse.headers.append(
      "x-snippets-hello",
      "Hello from Cloudflare Snippets"
    );

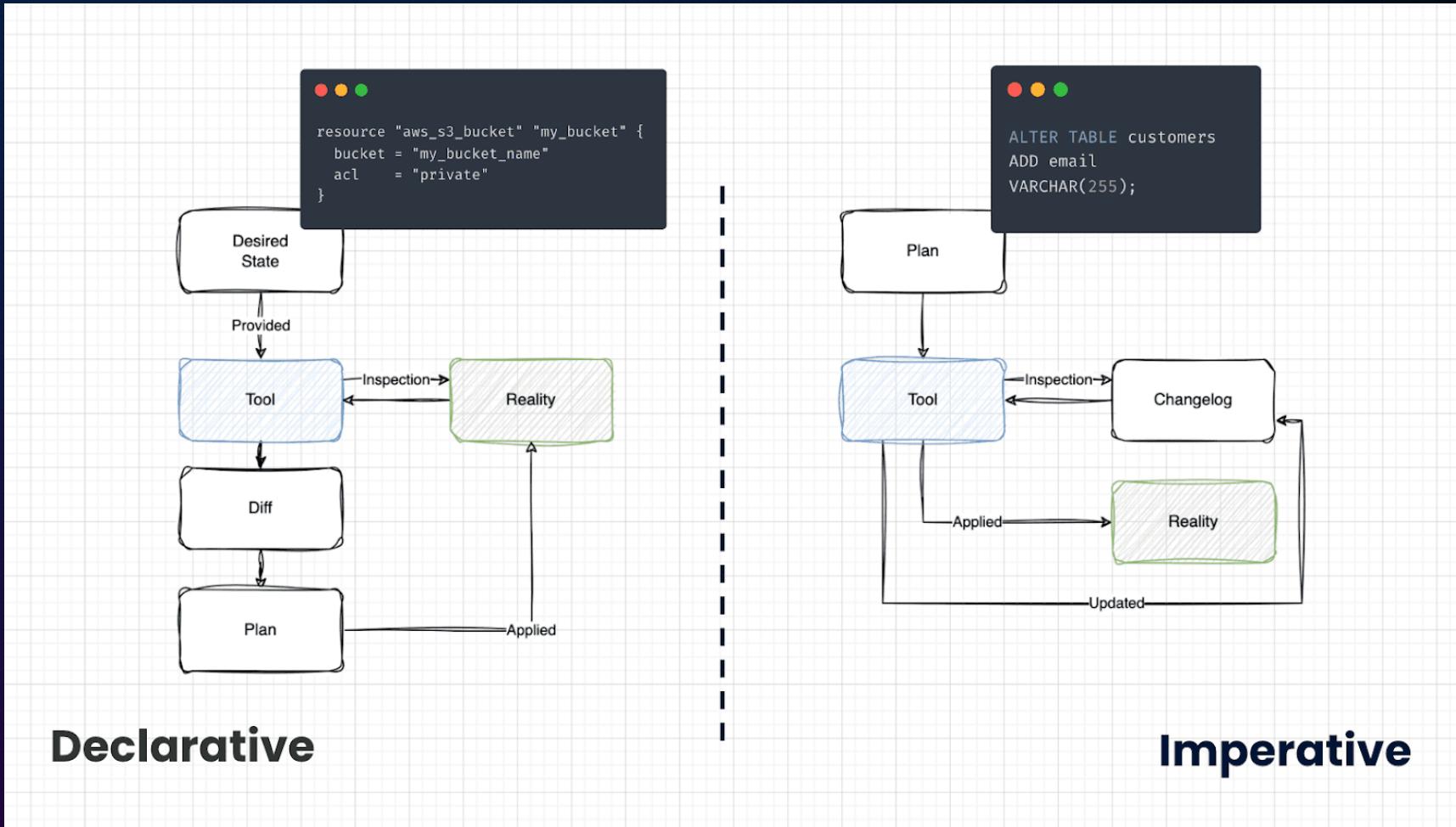
    // Delete headers
    newResponse.headers.delete("x-header-to-delete");
    newResponse.headers.delete("x-header2-to-delete");

    // Adjust the value for an existing header
    newResponse.headers.set("x-header-to-change",
"newValue");
    return newResponse;
  },
};
```



Schema as Code

JSON をもとに動的にスキーマ拡張



分散アーキテクチャとデータパイプライン

Apache BeamやFlinkのようなデータストリームパイプラインをCloudflareで実現可能

