



Edge Overview Series

Product Overview

Agenda

- API First
- Connected digital experiences
- Product line
- Edge
 - API services
 - Developer services
 - Analytics services
- API BaaS

The Core Design Principle - “API-First”

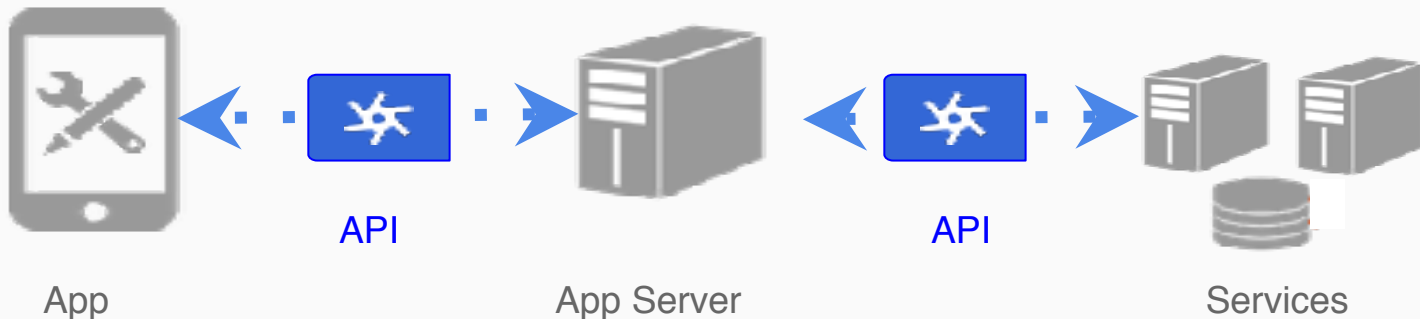
App Consumption

- API adaptations needed for apps
- Enable developers for business
- Security for app-to-API
- App and behavior analytics

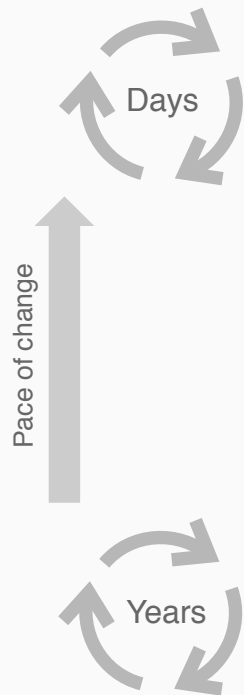
API Exposure

- APIs architected for abstraction
- Enable developers for API use
- Security for API-to-backend
- API Analytics

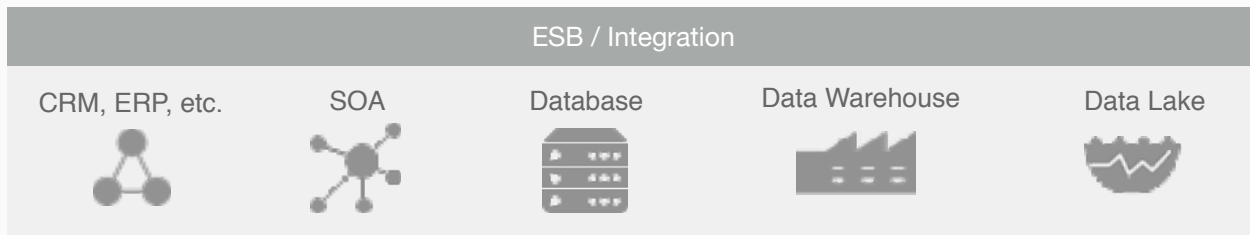
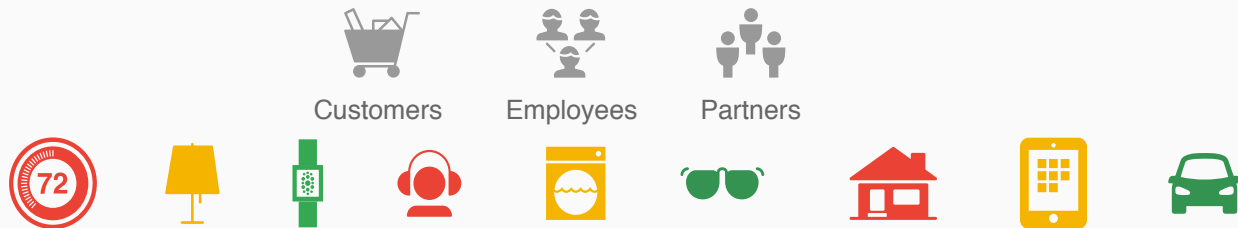
Analytics



The gap



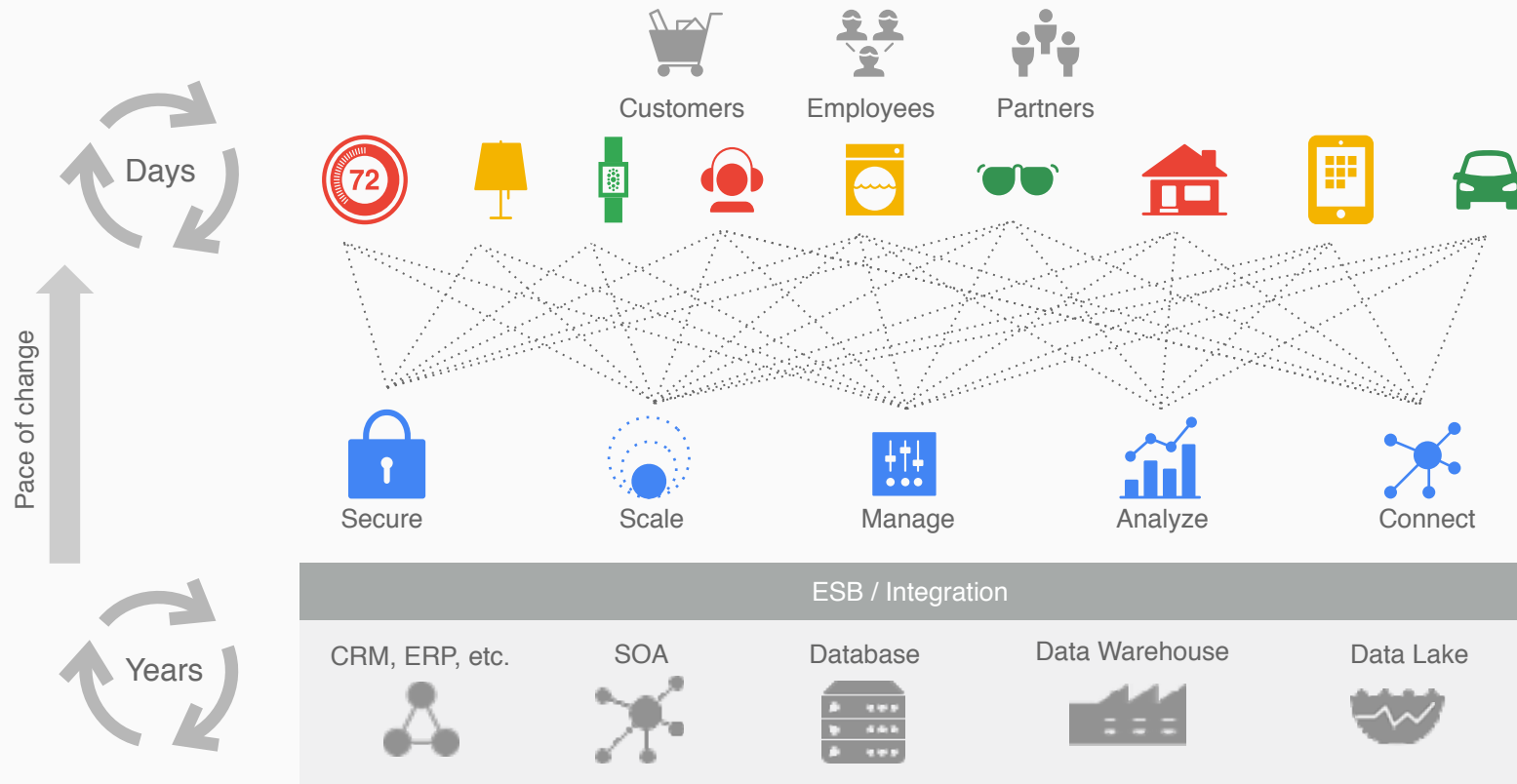
Connected Digital Experiences



Systems of Record

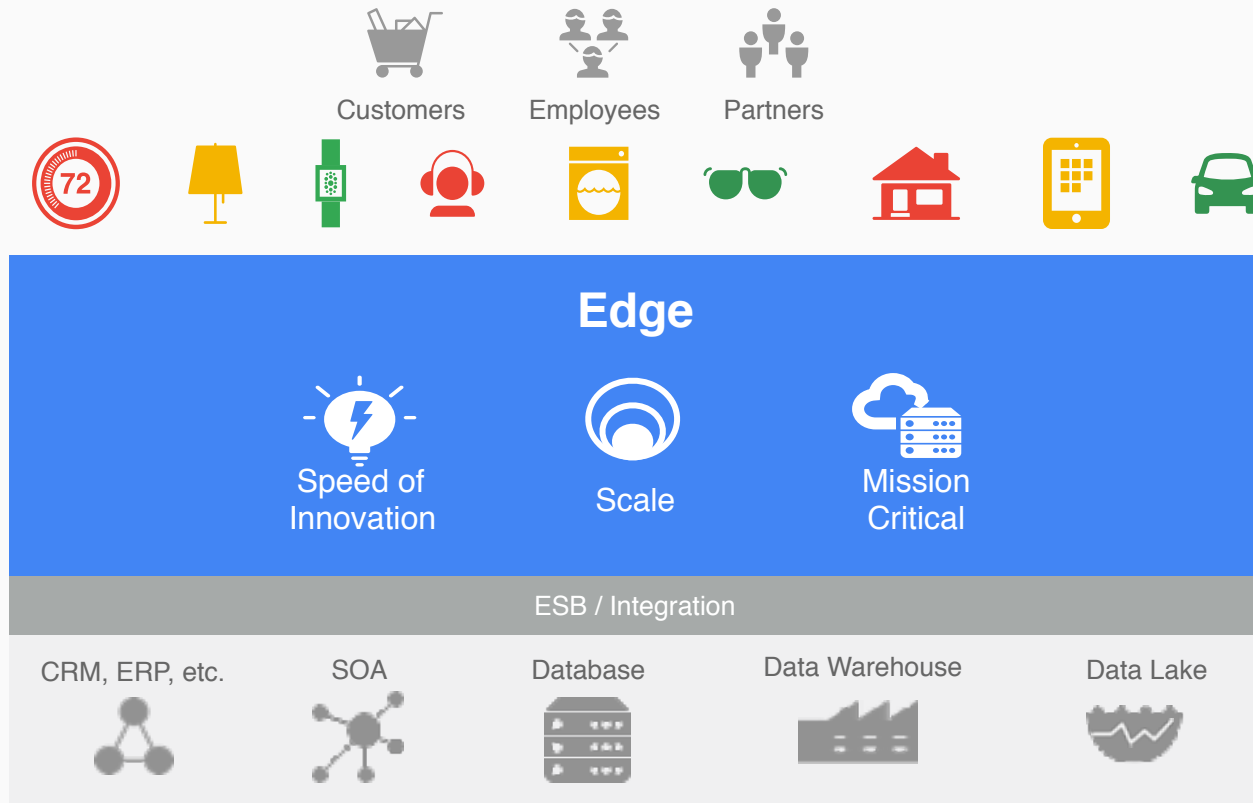
New demands

Connected Digital Experiences

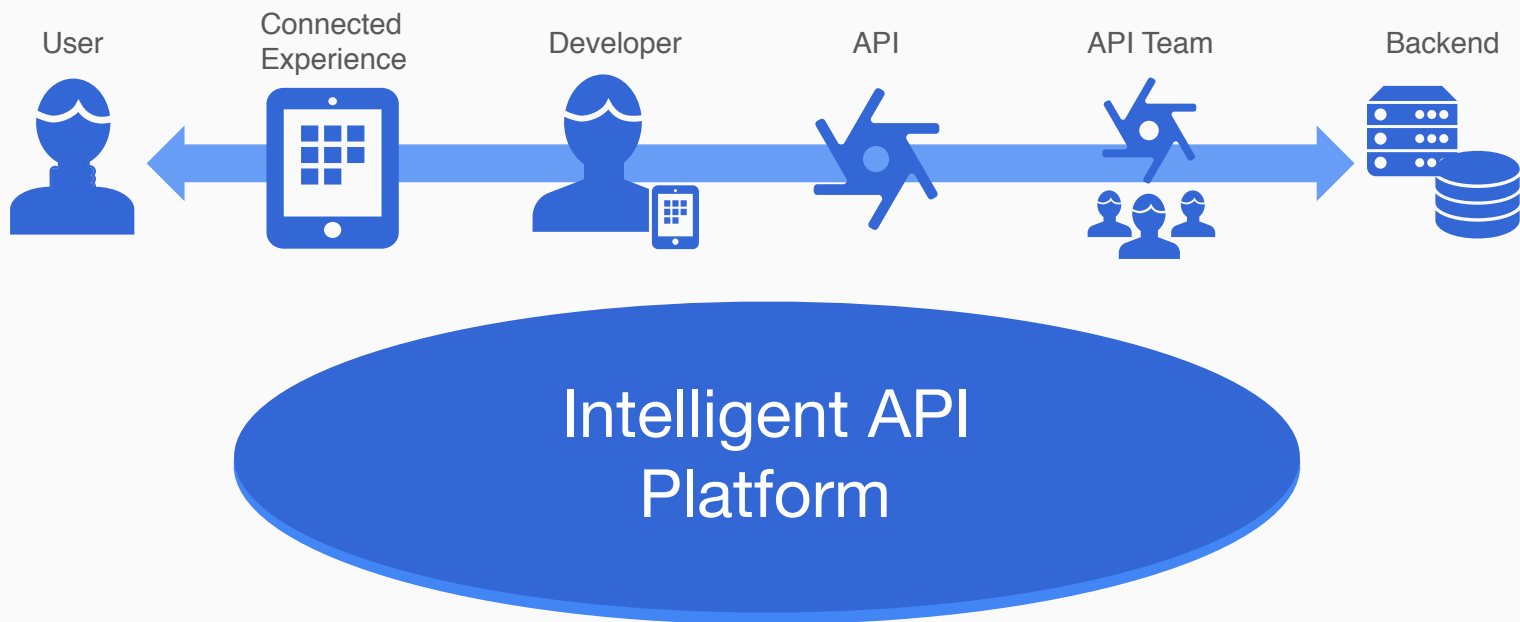


Bridging the gap

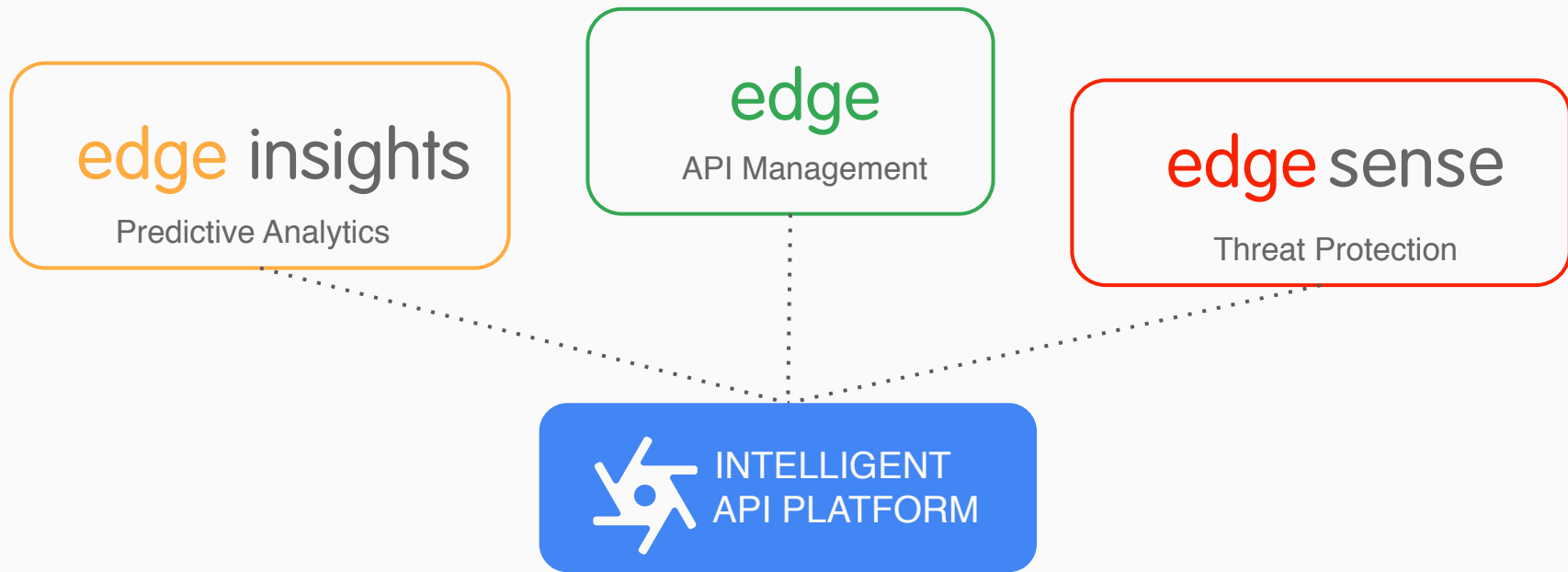
Connected Digital Experiences



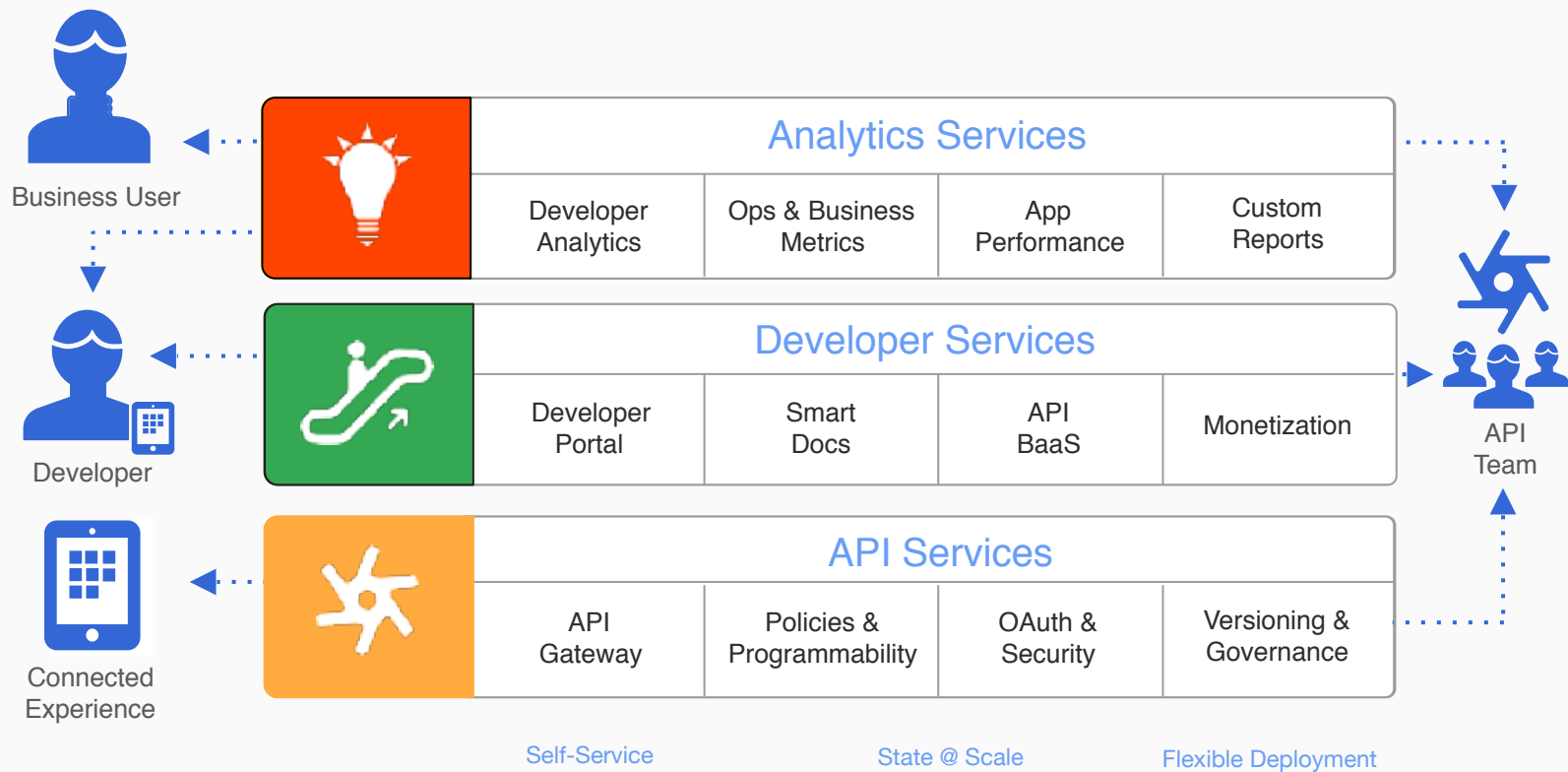
Intelligent API platform enables the Digital Value Chain



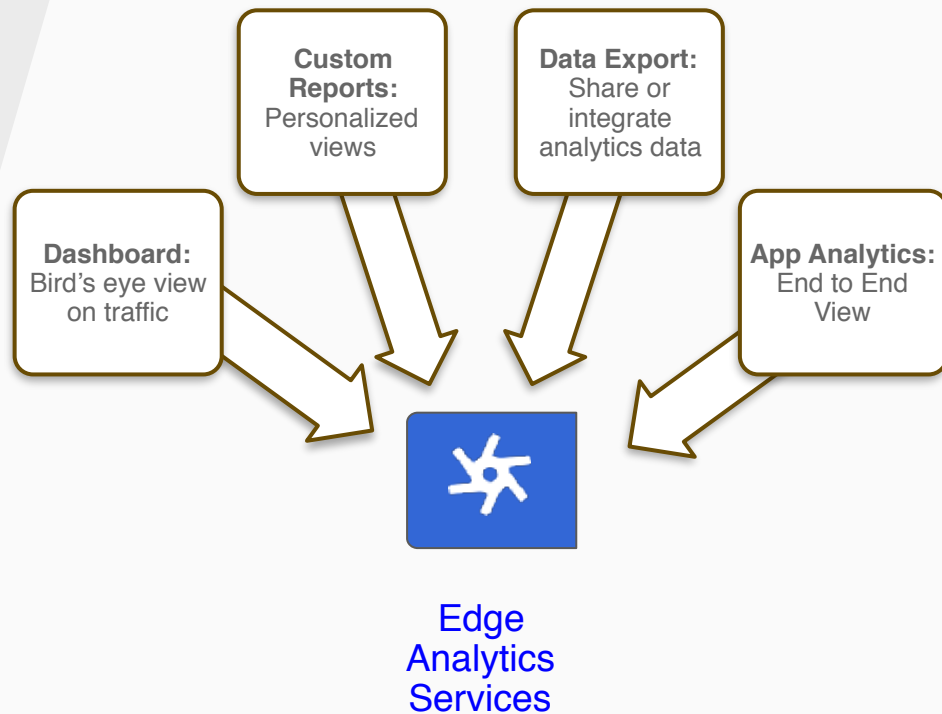
Public Cloud, Private Cloud & Hybrid



Edge - API Management



Analytics Services



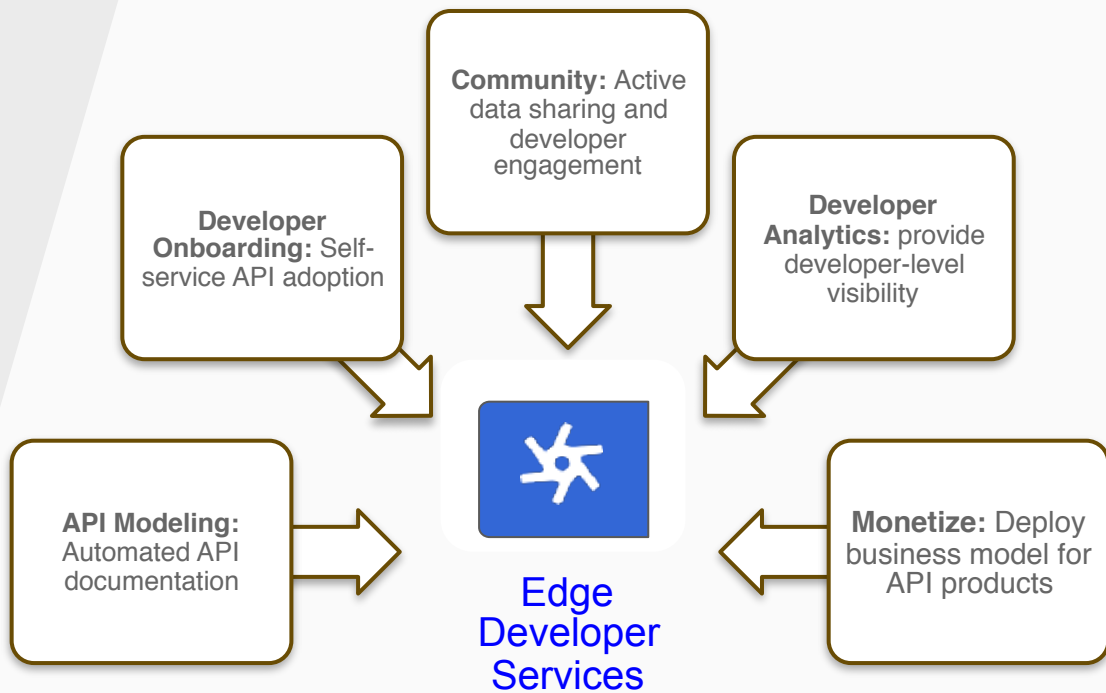
Analytics Services



Complete visibility from app end to backend.

Automatically and continuously collect all data about your API traffic out of the box.

Developer Services



Developer Services – Developer Portal

The screenshot shows the Payeezy Developer Portal interface. At the top, there's a navigation bar with the Payeezy logo, links for 'LINKS', 'ABOUT', 'Integration', 'Support', 'FAQs', and 'Blog'. On the right, there are buttons for 'CREATE ACCOUNT' and 'LOGIN'.

The main content area is titled 'Credit Card Payments' with a green 'info' icon. Below the title, a paragraph states: 'Use this method to submit payments credit and debit cards. Supported transaction types are 'authorization', 'purchase' and 'refund'.'

On the left side, there are three expandable sections: 'Resource URL', 'Header Parameters', and 'Try it out!'. The 'Try it out!' section is currently expanded, showing a tabbed interface with 'Authorize', 'Purchase', 'Refund', and 'Resume'. The 'Authorize' tab is active, displaying a green 'Authorize' heading and a paragraph: 'Use this to place a temporary authorization hold for the desired amount on the payer's statement. You can capture the authorized amount on completion of period or void/capture the transaction as required. Sample authorization payload for a credit card transaction is shown below.'

Below the text, there's a code block showing a JSON payload for an authorization request:

```
1 {
2   "merchant_ref": "Autooleasing-Sale",
3   "transaction_type": "authorization",
4   "method": "credit_card",
5   "amount": "100",
6   "currency_code": "USD",
7   "credit_card": {
8     "type": "visa",
9     "card_number": "4532015117890123",
10    "exp_date": "12/2018-12/2019",
11    "card_holder_name": "John Doe",
12    "billing_address": {
13      "street": "123 Main St",
14      "city": "New York",
15      "state": "NY",
16      "zip": "10001",
17      "country": "US"
18    }
19  }
20 }
```

On the right side, there's a 'Resource Summary' section. It includes a 'Category' of 'Make Payments' and a 'Notes' section stating: 'The below request payload shows "billing address". "merchant_ref", "card_ref" & "card_type" objects which are optional.'

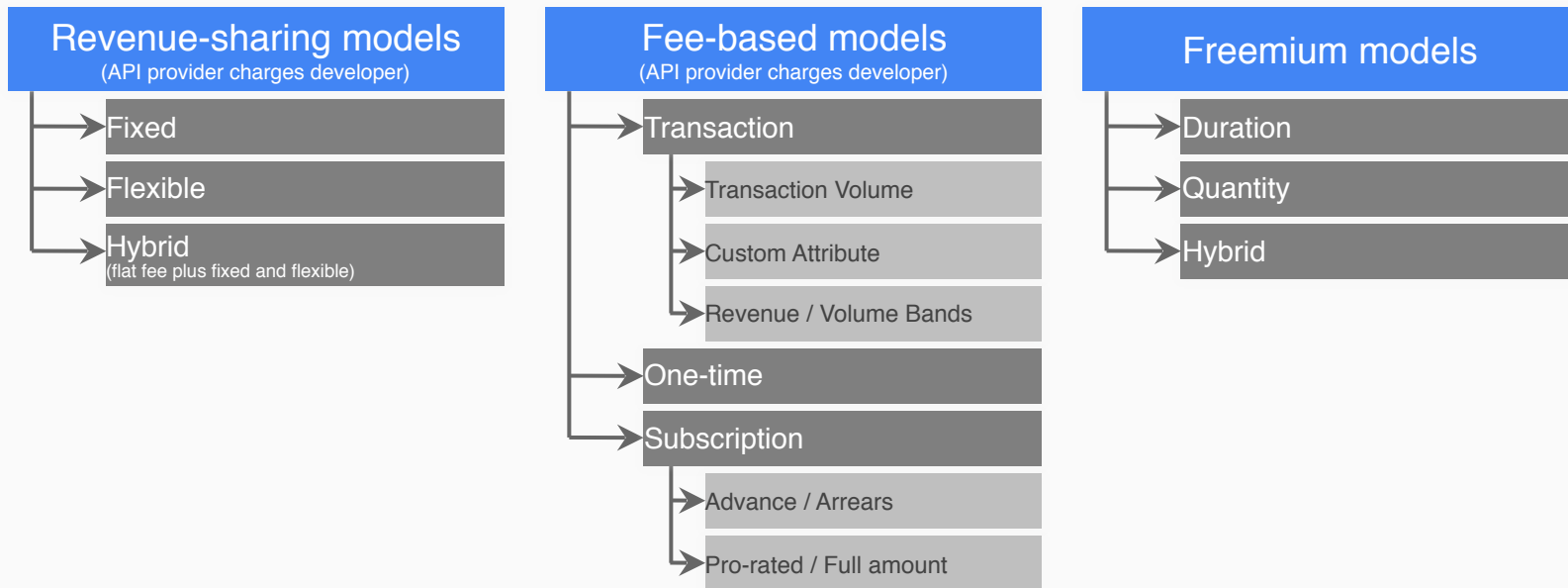
Below the notes, there's a link that says 'Click here' to learn more about how to make credit card payments. At the bottom right, there's a 'Request' tab with a JSON payload for a credit card transaction:

```
{
  "merchant_ref": "(string)",
  "transaction_type": "(string)",
  "method": "(string)",
  "amount": "(string)",
  "currency_code": "(string)",
  "credit_card": {
    "type": "(string)",
    "card_number": "(string)",
    "exp_date": "(string)",
    "card_holder_name": "(string)",
    "billing_address": {
      "street": "(string)",
      "city": "(string)",
      "state": "(string)",
      "zip": "(string)",
      "country": "(string)"
    }
  }
}
```

<https://developer.payeezy.com/>

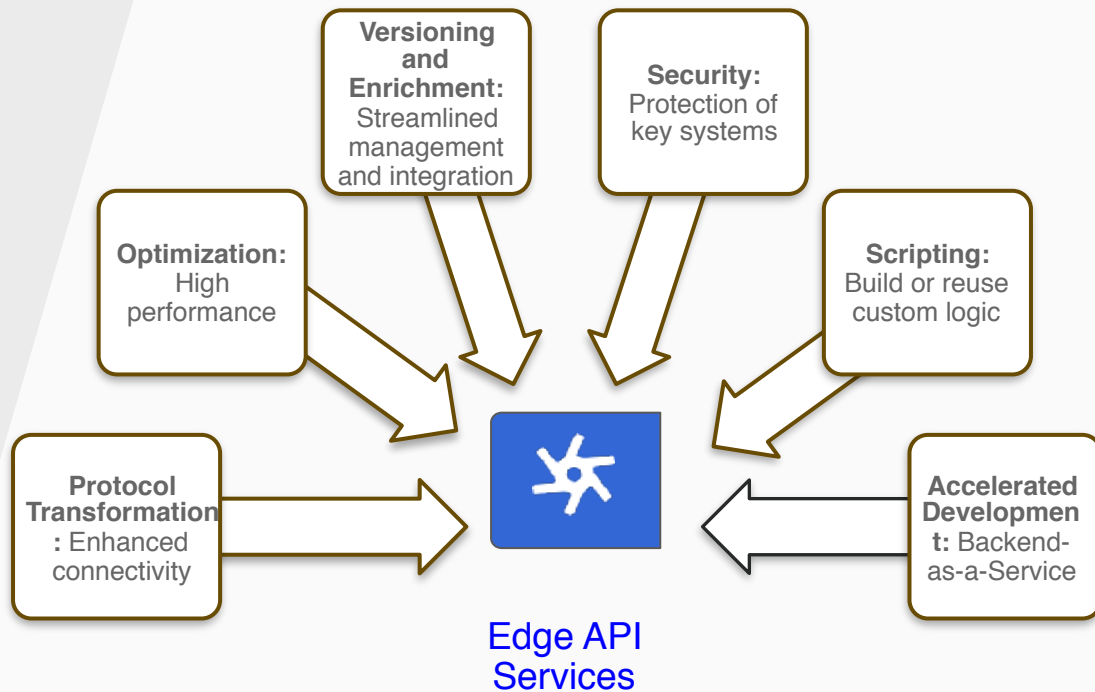
Streamline API adoption. Empower developers. Build an interactive community.

Developer Services – Monetization

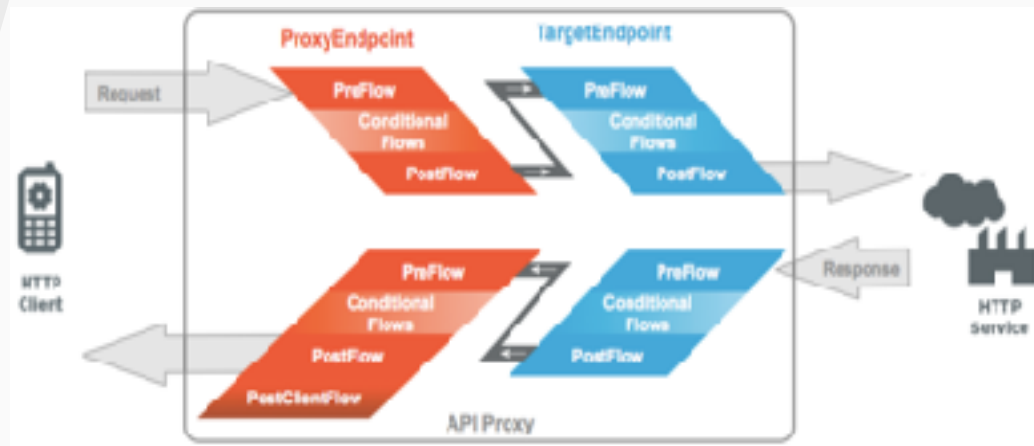


Flexible rate plans, internationalization support, usage tracking, limits and notifications

API Services



API Services



API Management | Security | API Programmability

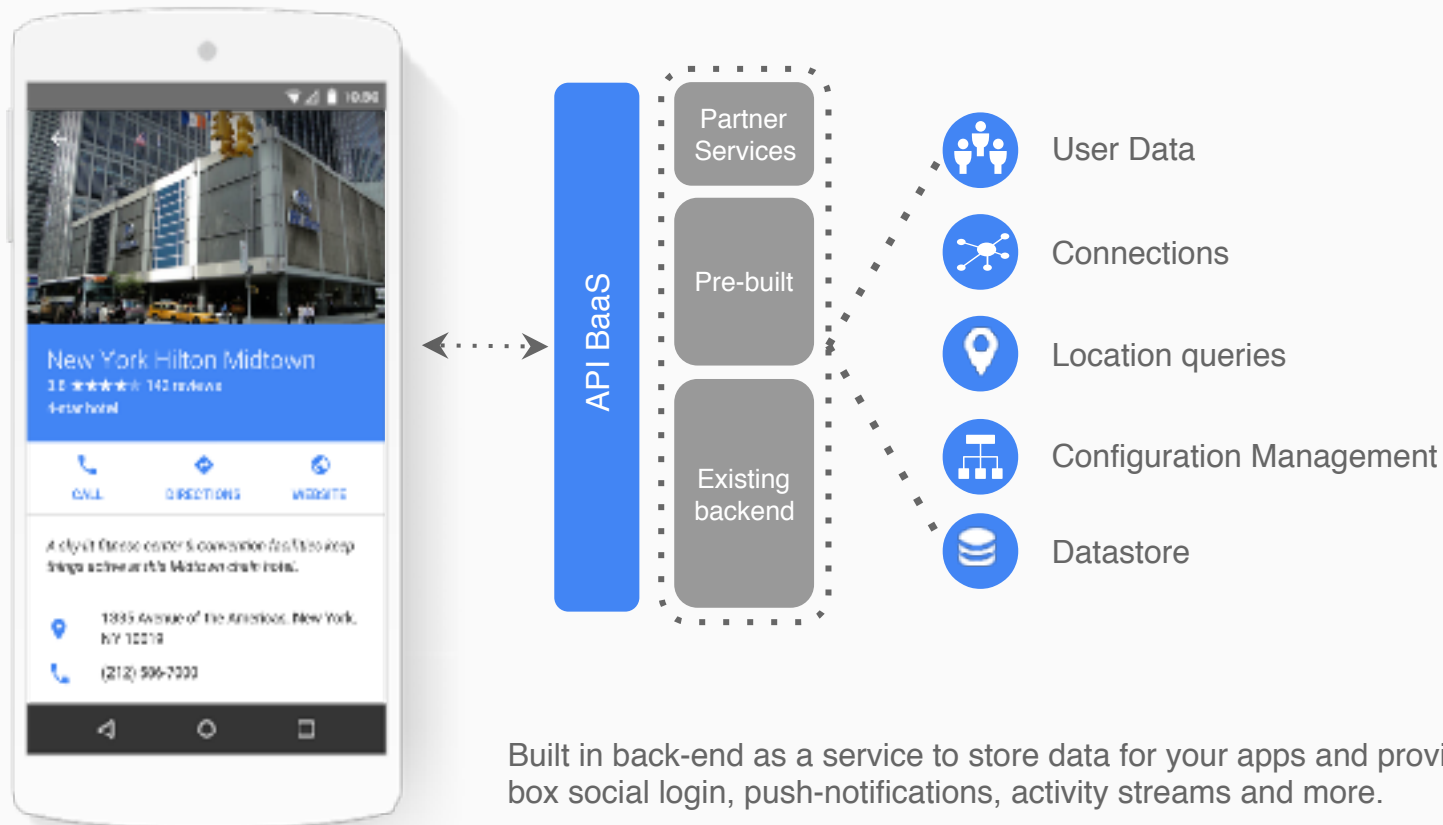
API Services - Policies

Traffic management policies	Mediation policies	Security policies	Extension policies
Traffic management policies let you configure cache, control traffic quotas and spikes, set concurrent rate limits, and so on.	Mediation policies let you perform message transformation, parsing, and validation, as well as raise faults and alerts.	Security policies let you control access to your APIs with OAuth, API key validation, and other threat protection features.	Extension policies let you provide custom policy functionality, with support for such features as service callout, message data collection, and calling Java, JavaScript, and Python behavior you have created.
<ul style="list-style-type: none">• Cache policies• Concurrent Rate Limit policy• Quota policy• Reset Quota policy• Spike Arrest policy	<ul style="list-style-type: none">• Access Entity policy• Assign Message policy• Extract Variables policy• JSON to XML policy• Key Value Map• Operations policy• Raise Fault policy• SOAP Message Validation policy• XML to JSON policy• XSL Transform policy	<ul style="list-style-type: none">• Access Control policy• Basic Authentication policy• JSON Threat Protection policy• LDAP policy *†• OAuth v2.0 policies• OAuth v1.0a policy• Regular Expression Protection policy• SAML Assertion policies• Verify API Key policy• XML Threat Protection policy	<ul style="list-style-type: none">• Java Callout policy *• JavaScript policy• Message Logging policy• Python Script policy *• Service Callout policy• Statistics Collector policy

* Cloud Enterprise only

† On-Premises installation only

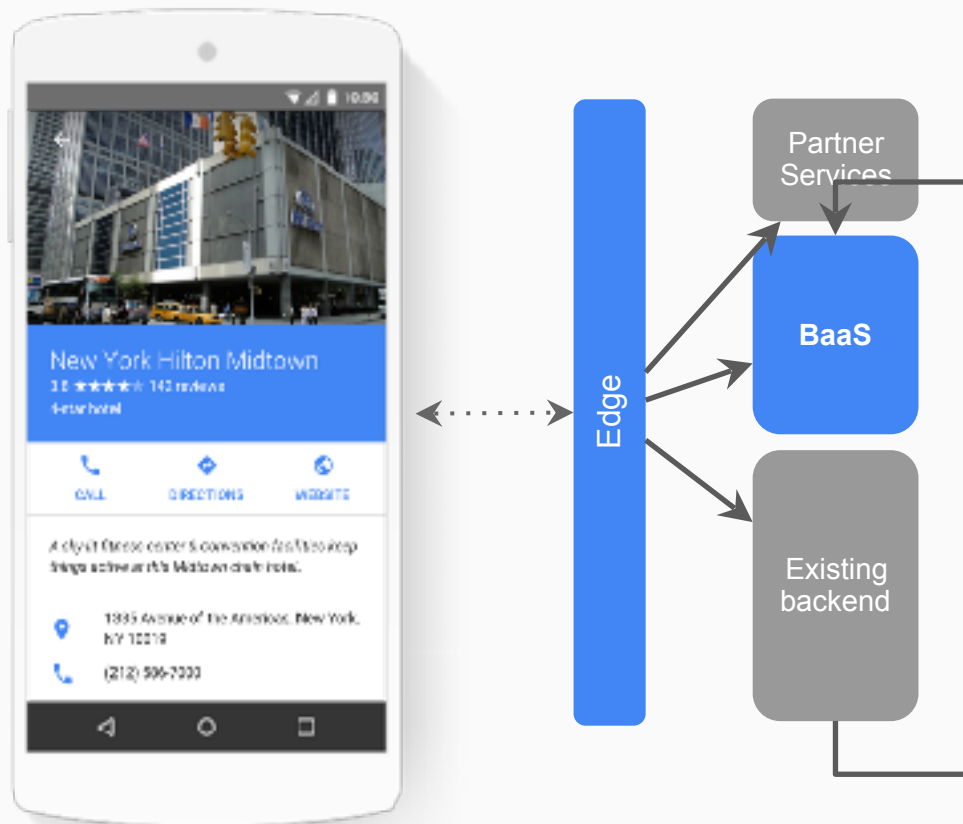
API BaaS – Capabilities



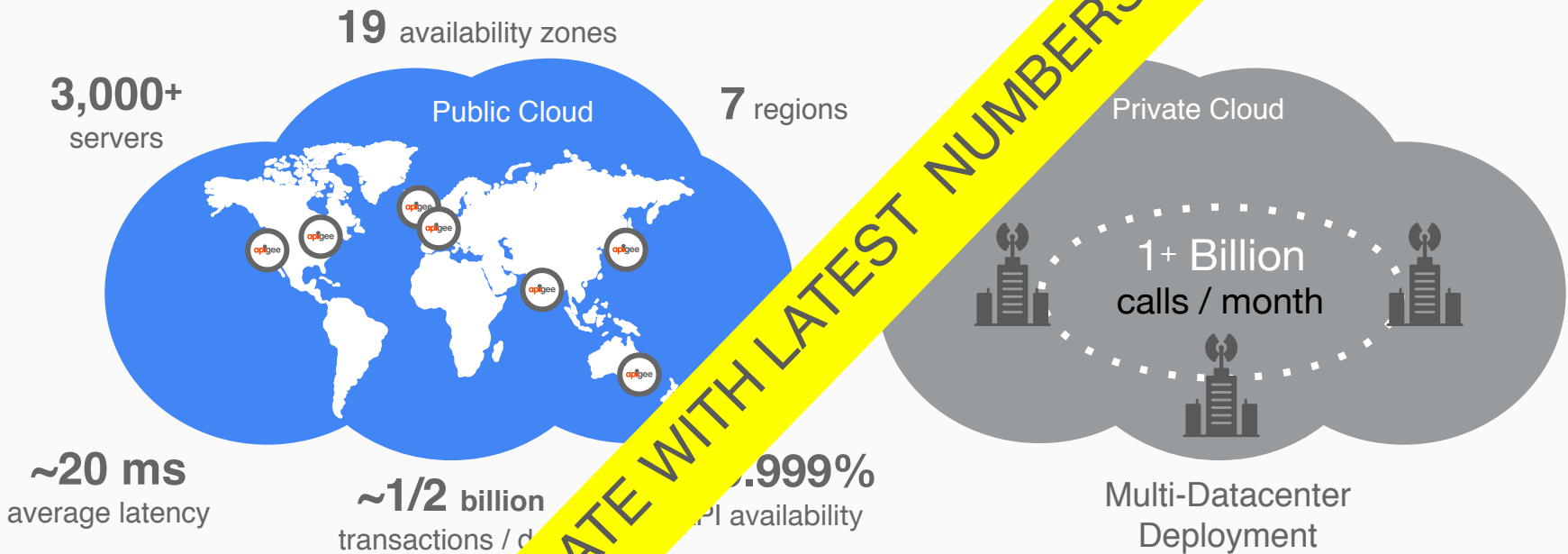
Built in back-end as a service to store data for your apps and provide out of box social login, push-notifications, activity streams and more.

API BaaS – Typical use cases

- Source or record for data generated by new API clients (Mobile, IoT, etc).
- Temporal data store for static or semi-static data
- Geolocation queries support.
- Push notifications.



Deploy in Public or Private Cloud



TODO: UPDATE WITH LATEST NUMBERS

Public Cloud = Private Cloud

Additional reading

[Why APIs?](#)

[The Definitive Guide to API Management](#)

[Two-Speed IT with APIs: Move Fast and Maintain Control](#)

THANK YOU