Building Your Own WAF as a Service and Forgetting about False Positives

Juan Berner

@89berner

Lead security developer
@Booking.com

Blog: medium.com/@89berner

Overview

- Introduction to WAF & deployment modes
- WAF as a service
- Blocking attacks without false positives or increased latency
- Demo

WAF?

- Web Application Firewall
- Mainly used to protect against Application Attacks
- SQLi, RCE, Protocol Violations, Rate Limiting ...

Deployment mode - Inline

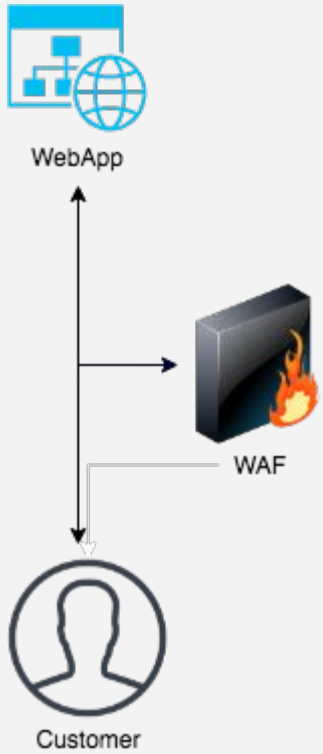
- **Pros:**
  - **Traffic inspection**
  - **Ability to block**
  - **Transparent for web servers**

- **Cons:**
  - **Network placement**
  - **Latency**

Deployment mode - Out of band

- **Pros:**
  - **Traffic inspection**
  - **Transparent for web servers**
  - **Simpler network placement**

- **Cons:**
  - **Can’t block attacks**
  - **PFS**



Deployment mode - Agent

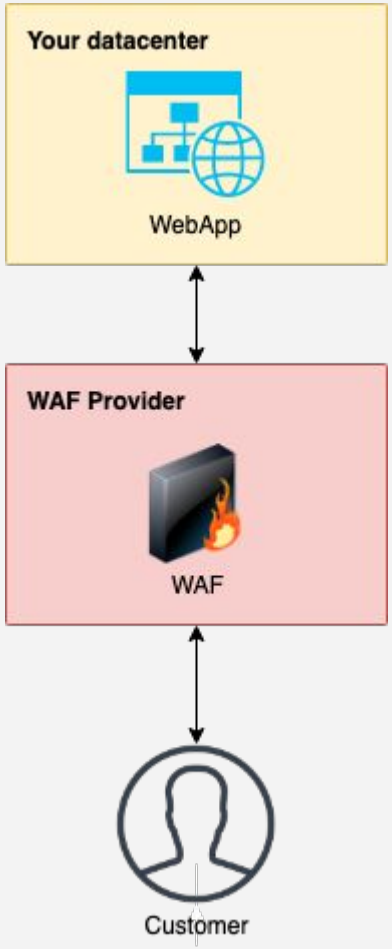
- **Pros:**
  - **Easier network placement**
  - **Simple to scale**

- **Cons:**
  - **More invasive on deployment environment**
  - **Can be less efficient on resource allocation**

Deployment mode - Cloud

- **Pros:**
  - **Simple to setup and scale**
  - **Network effect**

- **Cons:**
  - **Out of your control**
  - **Latency added**



Caveats with typical WAF Solutions

- Network placement
- Availability and performance concerns
- False positive rate
- Lack of control from developers

Building the WAF as a Service

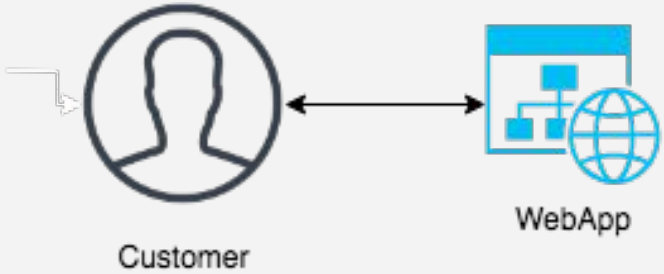
- Removes FP by having an understanding of the application context
- No need for an appliance, just add an API call
- Blocking behaviour is decided by the application
- Ability to avoid latency for regular users

How could you build one?

- Open source components already exist
- Creating a log processing pipeline
- Building a WAF API
- Library for logs and calling API

Case study: Web application

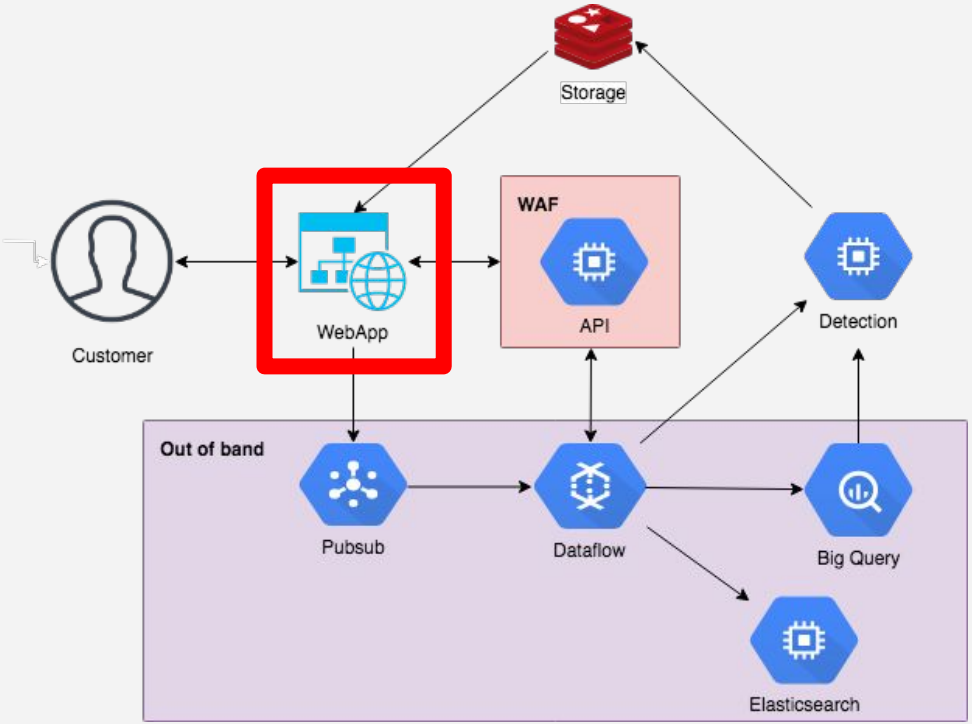
- Setup in Google Cloud
- Flask microframework
- Code available in github



Finding a middle ground

- Out of band mode removes concerns of latency added to users
- Inline mode provides security by blocking attacks
- Could we get the best of both worlds?

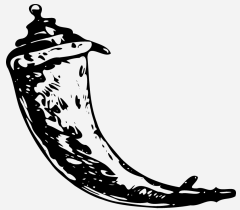
Components - Web application



Components - Web application

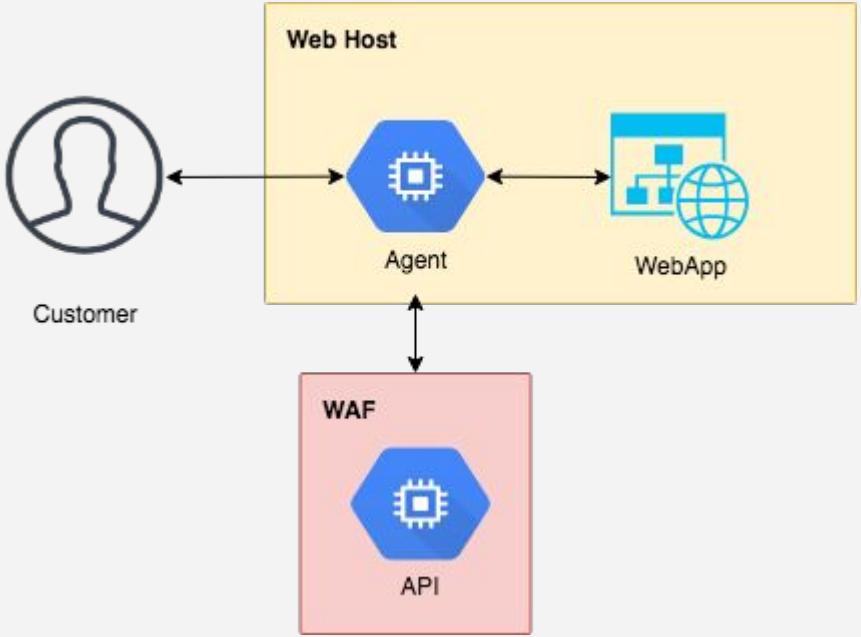
- Can decide which mode to work on
  - Inline
  - Out of band
- Sends logs with partial request data encrypted

Example: Flask



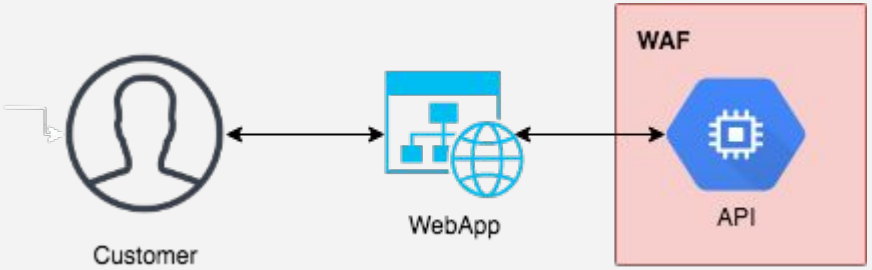
Components - Agent

- Acts as a reverse proxy
- Minimal footprint
- Application agnostic
- Can get settings from the application

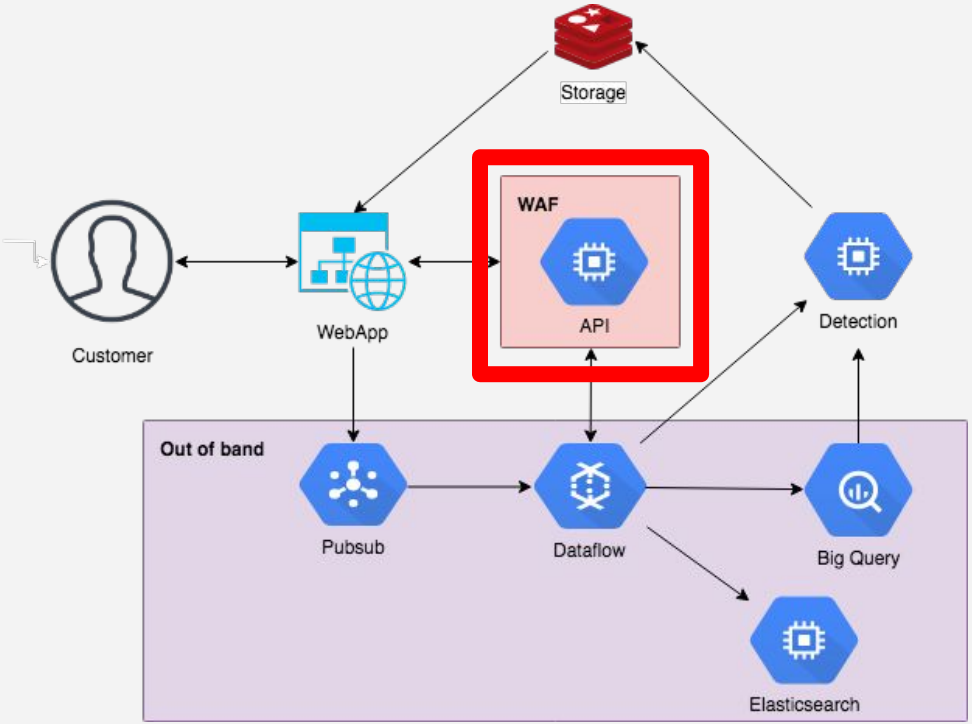


Components - Library

- Simple to implement
- Inherent risks
- Strategy for this talk



Components - WAF service



Components - WAF service

- Pluggable architecture
- Parallel nature of their components
- Applications can decide how to react

Components - WAF service

- Open source components
  - Modsecurity 

  - Naxsi

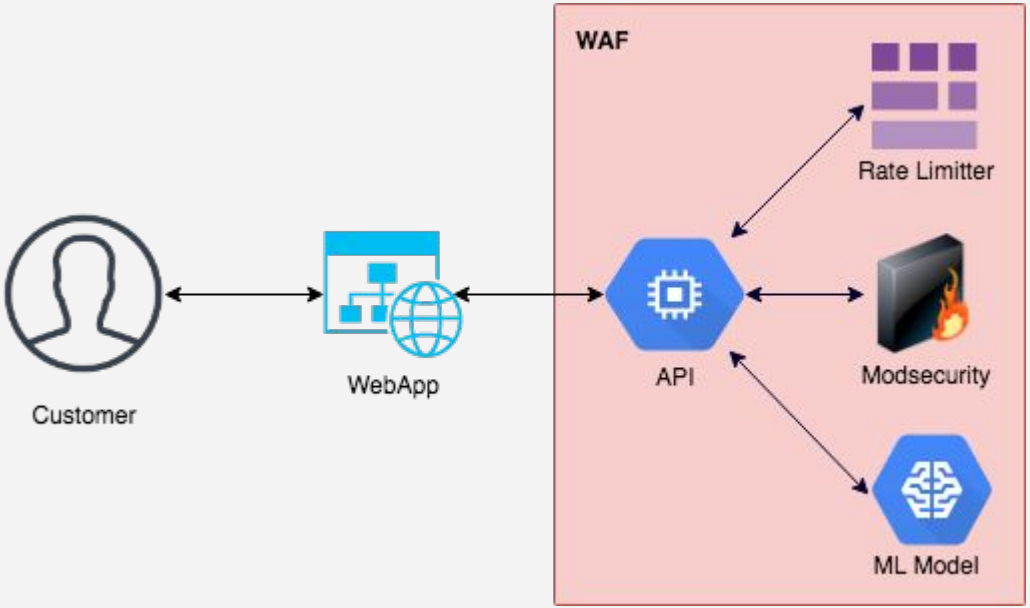
Components - WAF service

- Proprietary software or appliances
  - Reduced complexity of installation
  - Simple way of evaluation

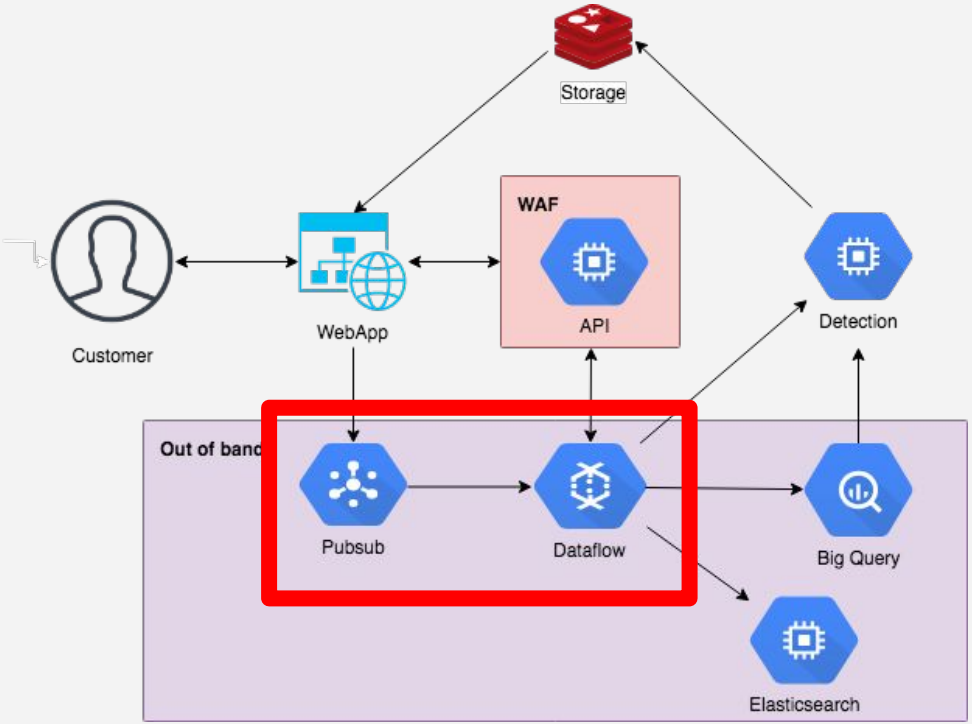
Components - WAF service

- Custom modules
  - Apply custom business logic
  - Implement simple services
    - Rate limiting
    - Rule engine for blocking
  - ML models

WAF service


WAF
Rate Limitter
Customer
WebApp
API
Modsecurity
ML Model

Components - Log processing

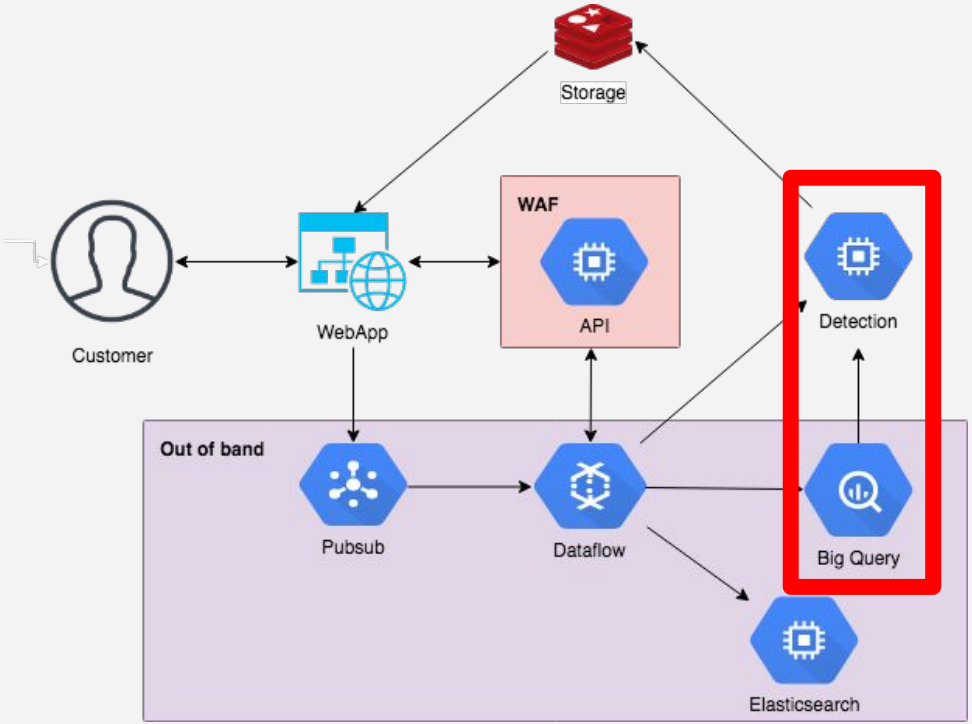


Components - Log processing

- Replays logs that were not in line against WAF
- Calculates scores through windows of time

Google Dataflow

Components - Detection

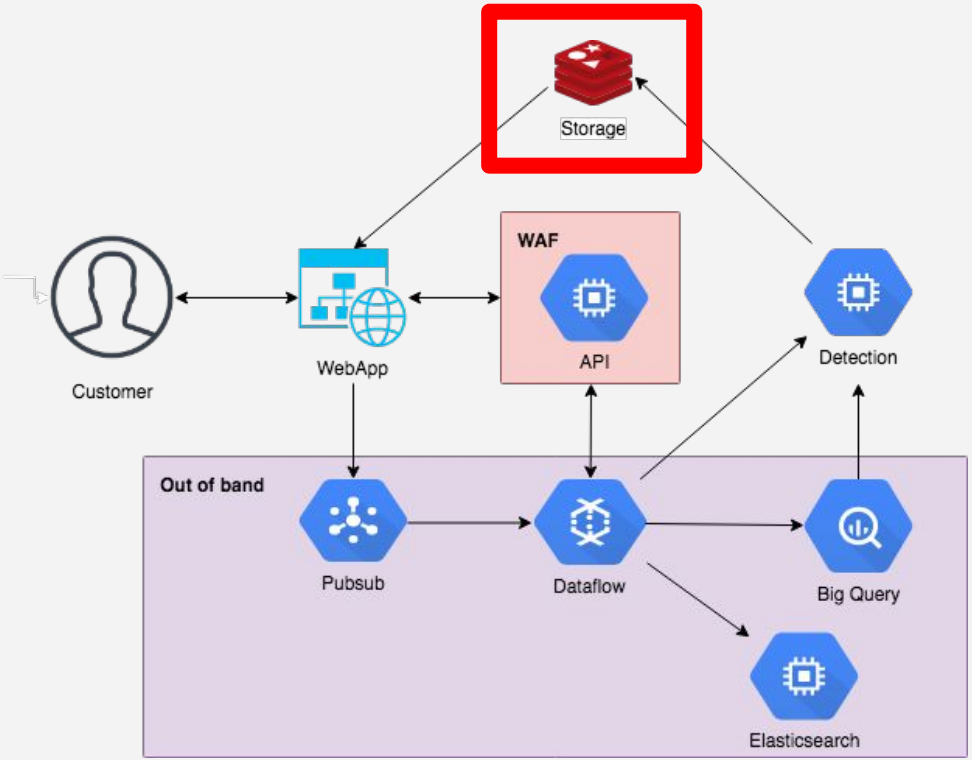


Components - Detection

- Triggered by Log Processing
- Business value
- Patterns of behaviour for FP

Google BigQuery

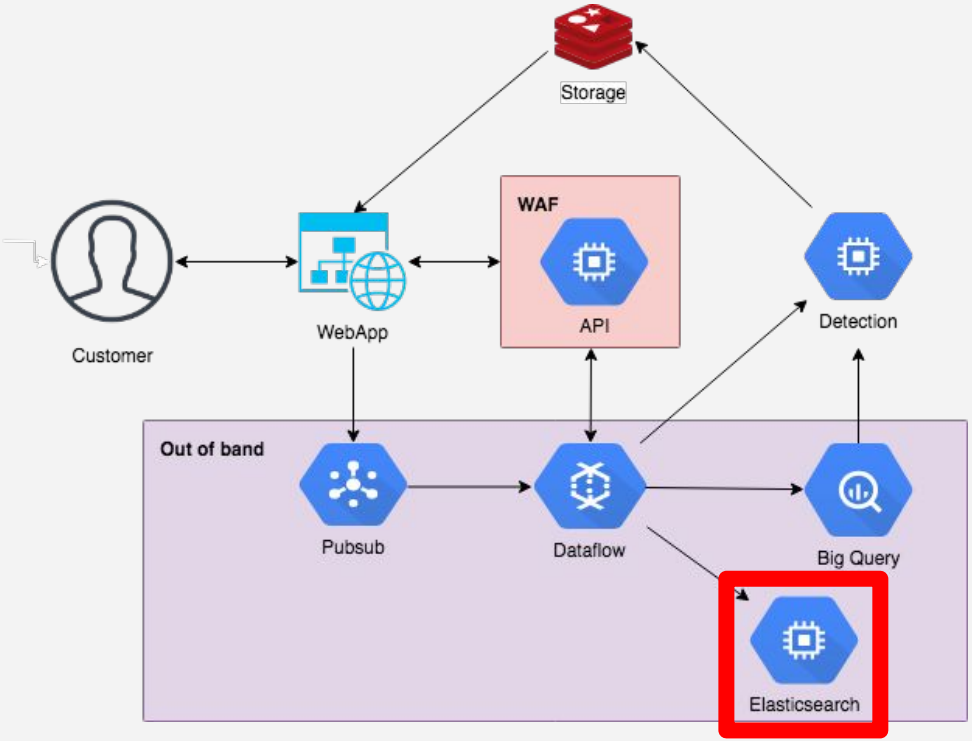
Components - State store



Components - State store

- Allows to store configuration
- Ideally fast lookup for caching

Components - Visualisation

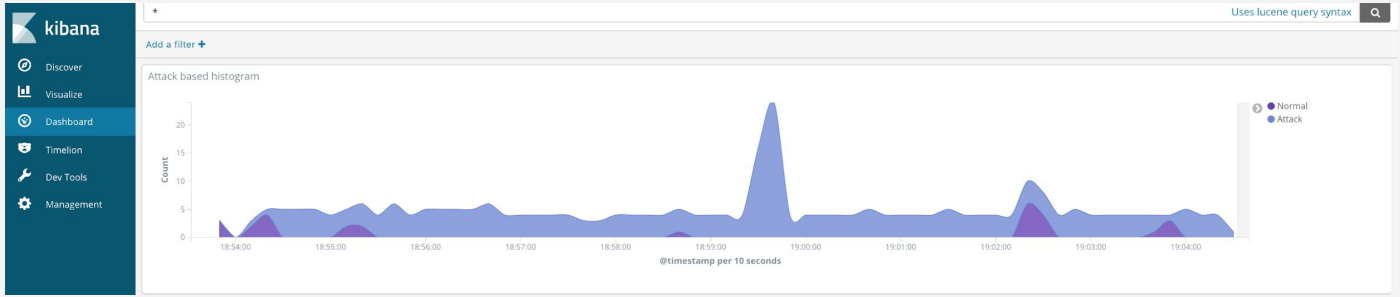


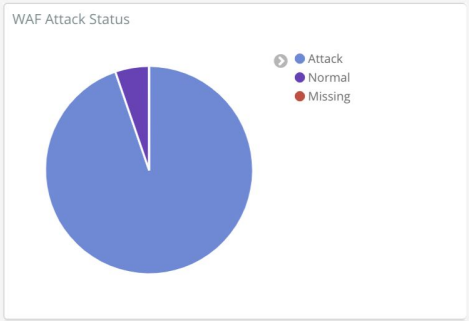
Components - Visualisation

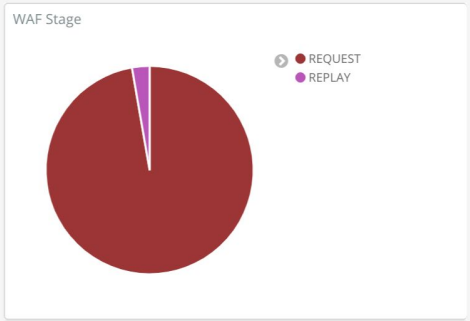
- Easily understand activity
- Visibility on attacks
- Performance metrics

Example: ELK

Components - Visualisation







Waf Alert Matched Vars

Matched Vars	Count
"SELECT * FROM INFORMATION_SCHEMA"	33
SELECT * FROM INFORMATION_SCHEMA	4
-1002%') OR 8810=9444 AND ('%'='	1
-1003 OR 9348=8785	1
-1008 OR 7376=7376#	1
-1010') OR 6010=6010-- -	1

Components - Visualisation

Waf search

1–50 of 311

Time	waf_request_time_spent	waf_mode	waf_request_answer.modsecurity.alerts.alert.logdata	waf_block	waf_status	waf_request_answer.rate_limiter.is_at
November 29th 2018, 19:04:30.882	0.4041290283203125	identifier	Matched Data: s)&(s found within ARGS:id: 1') OR NOT 5749=6432 AND ('YQom' LIKE 'YQom	REQUEST	Attack	0
November 29th 2018, 19:04:28.703	0.3751790523529053	identifier	Matched Data: s&sos found within ARGS:id: 1' OR NOT 6486=6486 AND 'vqoO'='vqoO	REQUEST	Attack	0

#	waf_request_answer.modsecurity.is_attack	1
#	waf_request_answer.rate_limiter.is_attack	0
#	waf_request_answer.rule_engine.is_attack	0
t	waf_request_answer.status	Attack

Components - Management

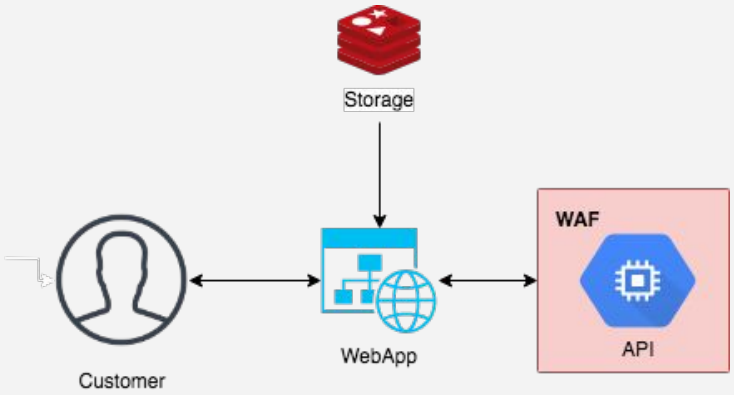
```
root@waf-5c65c789c8-sxzr6:/# python /api/manage.py --show-config=1
========================================================================
WAF Configuration
+--------------------+-------------+
|       Config       |   Status    |
+--------------------+-------------+
|   Request Stage    | identifier  |
|   Response Stage   |  disabled   |
| Waf Proxy Routing  |  disabled   |
| Scoring threshold  |      5      |
+--------------------+-------------+
========================================================================
WAF Identifier based routing:
+-------------+-----------------+---------------------+-----------------+
| Identifier  |      Value      |       Added at      |   Created by    |
+-------------+-----------------+---------------------+-----------------+
|     ip      |    10.40.1.8    | 2018-11-29 22:45:23 | alerter_script  |
|     ip      | 62.140.137.152  | 2018-11-29 22:45:26 | alerter_script  |
+-------------+-----------------+---------------------+-----------------+
========================================================================
Virtual patching routing:
+-----------+---------------------+
| Endpoint  |       Added at      |
+-----------+---------------------+
|   ping    | 2018-11-29 22:49:58 |
+-----------+---------------------+
========================================================================
Block Rules Configured:
+-------------+-------------------------------------------------------------------------------------------+---------------------+----------------+
| Identifier  |                                           Value                                           |       Added at      |   Created by   |
+-------------+-------------------------------------------------------------------------------------------+---------------------+----------------+
| user-agent  | mozilla/5.0 (windows nt 6.1; win64; x64; rv:58.0) gecko/20100101 firefox/58.0             | 2018-11-29 22:49:44 | manage_script  |
+-------------+-------------------------------------------------------------------------------------------+---------------------+----------------+
========================================================================
Rate limit counters:
+---------------------+-------------+---------------+---------+-----+
|     Time Bucket     | Identifier  |     Value     | Counter | TTL |
+---------------------+-------------+---------------+---------+-----+
| 2018-11-29 22:49:00 |     ip      |   10.40.1.8   |    50   |  55 |
| 2018-11-29 22:49:00 | user-agent  | notsequelmap  |    50   |  55 |
| 2018-11-29 22:50:00 |     ip      |   10.40.1.8   |    3    |  59 |
| 2018-11-29 22:50:00 | user-agent  | notsequelmap  |    3    |  59 |
+---------------------+-------------+---------------+---------+-----+
root@waf-5c65c789c8-sxzr6:/#
```

How to block?

- Detection decides when to send traffic to the WAF
- Can also be triggered manually

Traffic routing

- Fingerprint based routing
  - Blocks based on scores
  - IP, client_id, combinations, 0day signatures ..
  - Added automatically or manually



Traffic routing

- Net block based routing
  - ISP
  - Hosting providers
  - Tor exit nodes / Proxies

Traffic routing

- Virtual Patching
  - Always route particular vulnerable endpoints
  - Select for combination of parameters if needed
  - Example: website.com/?*vuln_param*=

FP rate management

- Detection FP vs blocking FP
- Key to allow blocking without impacting users
- Acceptable rate might change per application
- Tuning can become unbearable in highly changing applications

FP rate management

- Business logic
  - How trustworthy is a user/ip?
  - Key business activity
  - What would be the impact on blocking them

FP rate management

- Historical Analysis
  - How normal is this type of request for this endpoint?
  - How does this user compare with others
  - How common are detection FP in this endpoint

FP rate management

- Context analysis
  - How many times have they triggered a FP
  - How many requests have they sent

FP rate management

- **Example: Sleep(**
  - ***message="I will sleep(1 or 2 days)"***
    - **Might be detected as SQLI**
    - **Probability of FP is independent from each other**

FP rate management

- Independant SQLI FP rate: 0.1%
- Our aim, 0.00001% (0.01^5)
- Score needed => 5 * Reputation Score
- Aimed at attacks that need volume

Hybrid mode

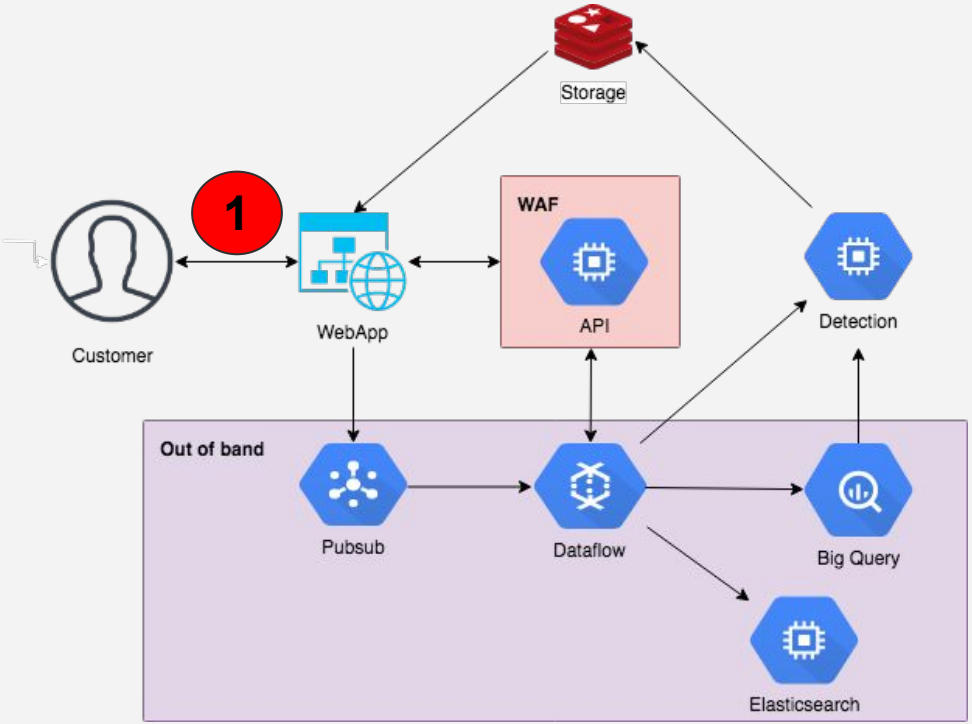
- Benefits
  - WAF does not add latency for good users
  - Flexibility
  - Removes FP’s

Hybrid mode

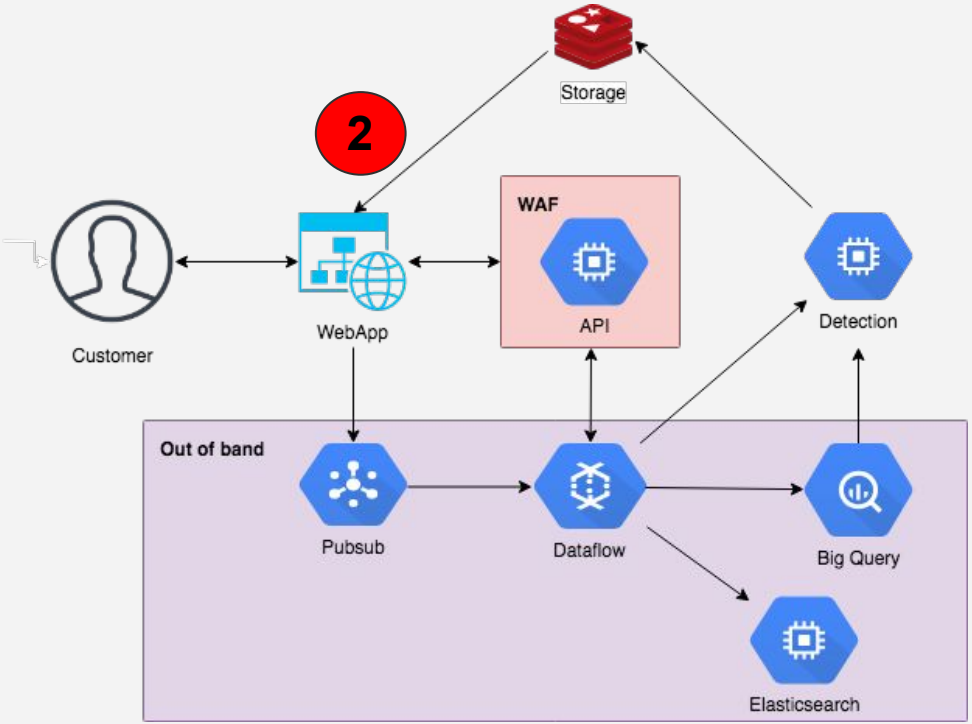
- Caveats
  - Delayed response time for blocking when using identifier mode
  - Increased complexity


ONE DOES NOT SIMPLY
PERFORM A LIVE DEMO
memegenerator.net

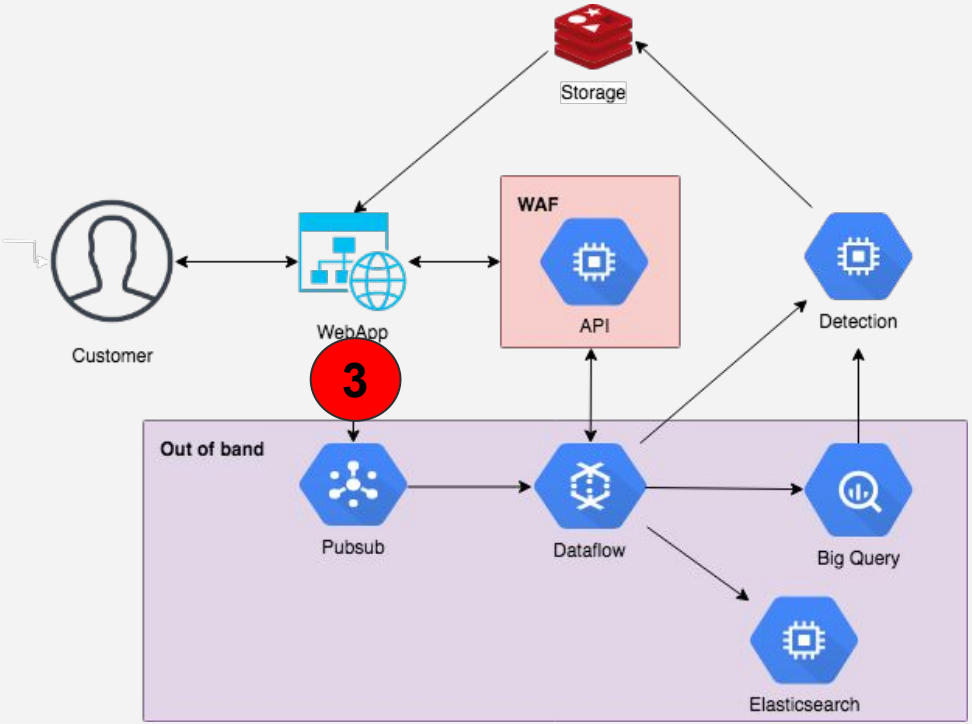
Request lifetime - Initial request



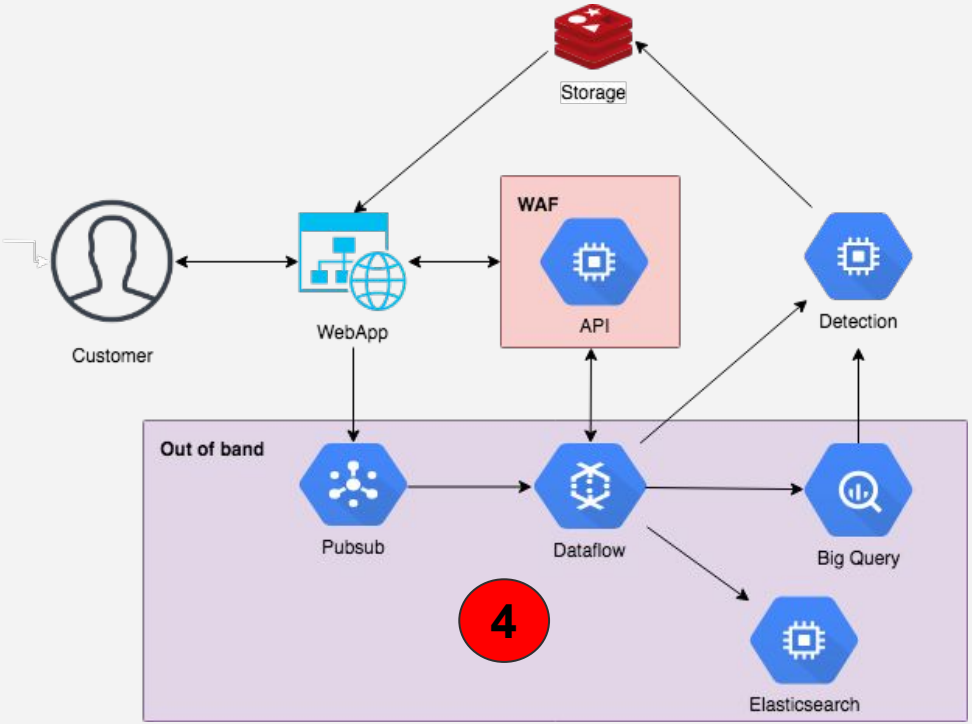
Request lifetime - Cache check



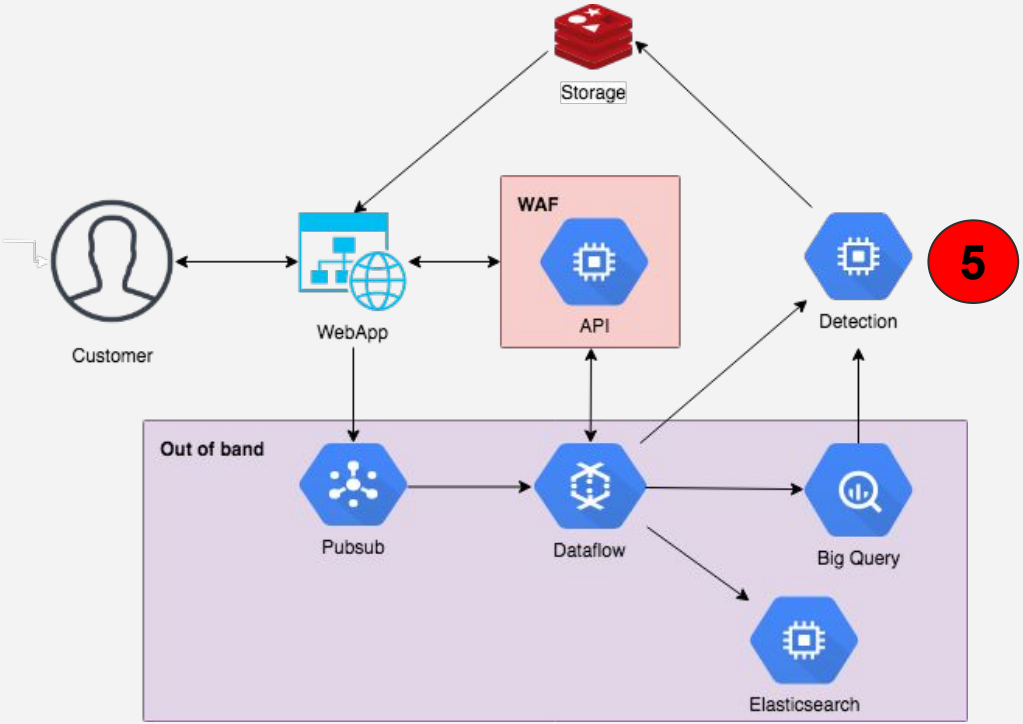
Request lifetime - Request encapsulation



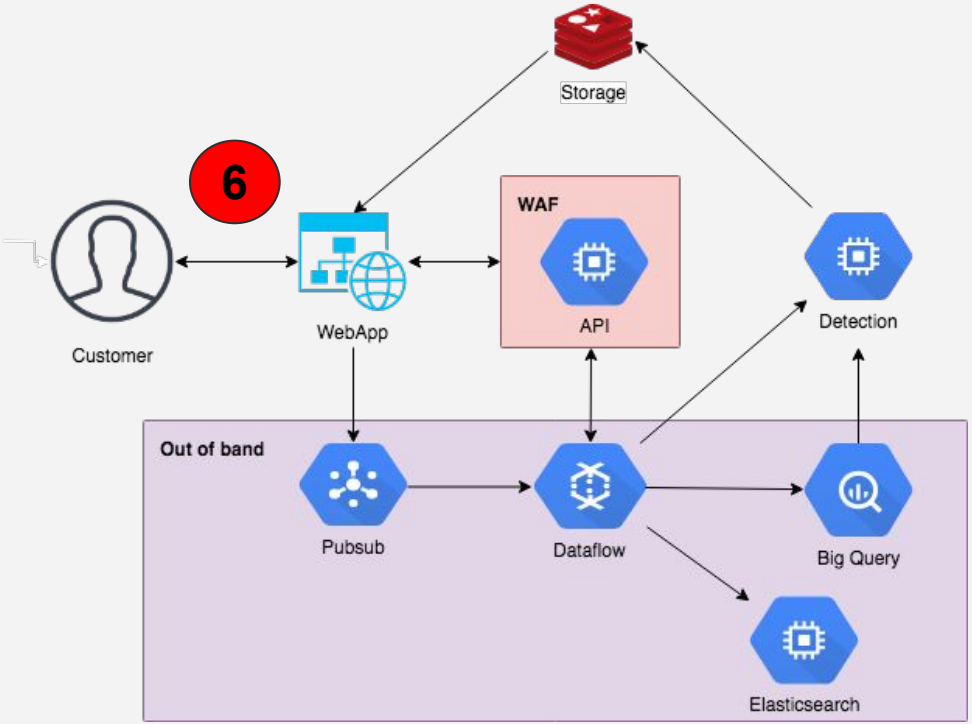
Request lifetime - Out of band processing



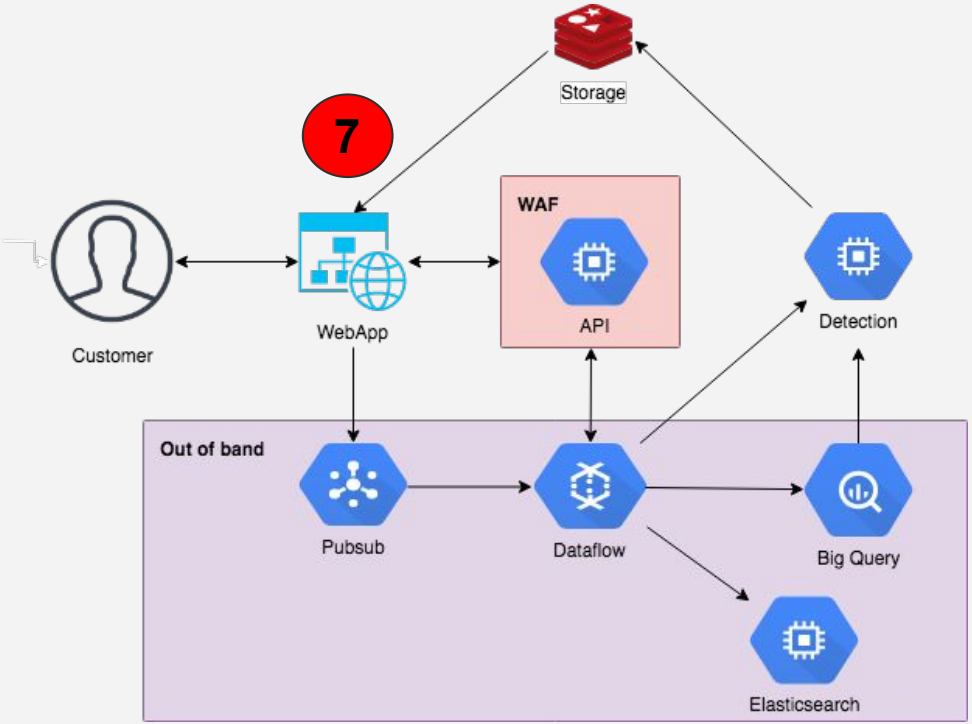
Request lifetime - Attack detection



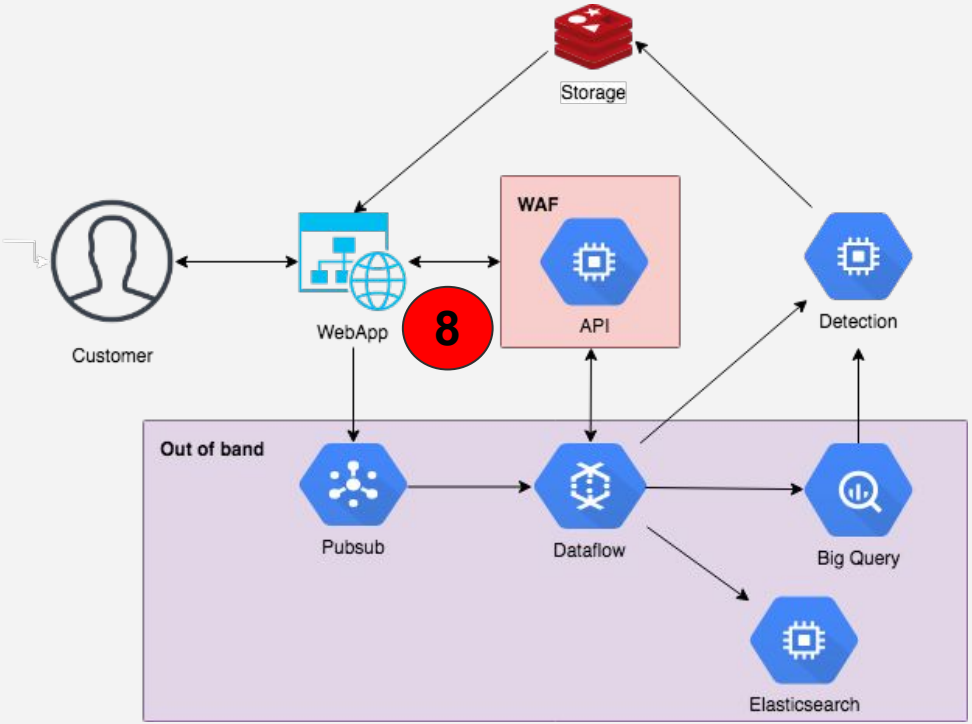
Request lifetime - Additional requests



Request lifetime - Cache update



Request lifetime - Inline behaviour



Summary (1/2)

- Reduced customer impact
  - Use hybrid mode to only add latency to malicious actors
  - Stops false positives from affecting customers through understanding history, context and business metrics
  - Specify different behaviour based on endpoint's risk

Summary (2/2)

- **Flexibility**
  - **Extensible through third party products or custom plugins**
  - **Allows developers to integrate through api calls where needed**

What now?

- **Try it!**
- **https://github.com/89berner/waf-api-talk**
- ***git clone https://github.com/89berner/waf-api-talk && cd waf-api-talk; ./setup $YOUR_GCP_PROJECT***
- **Questions?**