

Colibri astro-colibri.web.app/#/

select action Latest transients Cone search

personalize status: test

filter latest transients since: 2020-05-04 until: 2020-11-03 Fermi Swift Icecube

selected ra: 144.03° dec: -43.32° error: 6.23° time: 07:35:38 date: 10/12/2020 cone search

show ra: -72.61° dec: 57.04° error: 0.05° time: 05:46:30 date: 10/13/2020 cone search

show ra: 64.84° dec: -15.82° error: 0.05° time: 00:06:22 date: 10/13/2020 cone search

show ra: 144.03° dec: -43.32° error: 6.23° time: 07:35:38 date: 10/12/2020 cone search

show ra: 107.95° dec: -2.39° error: 4.95° time: 22:04:36 date: 10/11/2020 cone search

show ra: 104.32° dec: -3° error: 0.05° time: 00:06:22 date: 10/13/2020 cone search

altitude (deg) vs hours from UTC midnight graph

Detailed info about selected source:
 ivom: live/raissa_gfsc/gcn/Fermi/CGBM_Grid_Pos_2020-10-12T05:35:38.50_624173743_0-482
 observatory: Fermi

Detailed monthly visibility graph



TS2020 March 2021 Meeting

ASTRO-COLIBRI astro-colibri.com

The coincidence library for real-time inquiry for multi-messenger astrophysics

Patrick Reichherzer on behalf of the COLIBRI development team



Section

Motivation for Astro-COLIBRI

Motivation for Astro-COLIBRI

- Flares of known stable and transient sources can occur on different timescales
- Improve multi-messenger/wavelength follow-up:
 - Quickly acquiring an overview over both stable sources and transient events in the relevant phase space

What we need

- Automatically collect
 - Archival data
 - Transient activity data over various timescales
 - Filter and combine VoEvents of same events and combine with GCN Circulars
 - Summarize it in human and machine-readable formats
- Provide an (interactive) graphical representation
 - Multi-wavelength and multi-messenger data
 - Filtering in space & time
- Correlate transient alerts automatically
- Connect to alert reception & link to real-time analyses



Motivation for Astro-COLIBRI

Challenges in the real-time analysis of the transient sky

- Many processes are not automatized in observatories
 - Error-prone
 - Time consuming
 - Incomplete

- Decision processes for schedule updates and alert issues may need to be taken by non-experts (e.g. tired shifters)

→ Need for automated access to all necessary information for prompt prioritization



Section

History of Astro-COLIBRI

History of Colibri – v0



H.E.S.S. collaboration meeting (10.2019) [link](#)

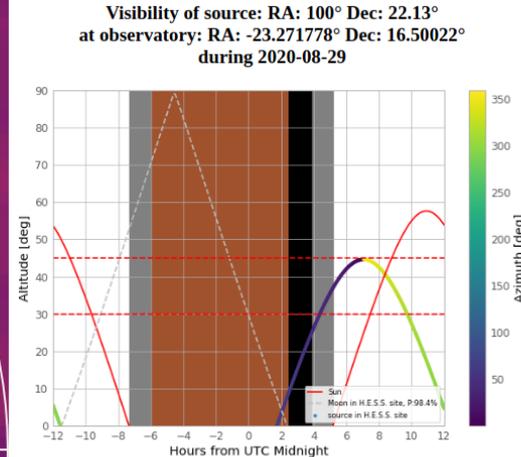
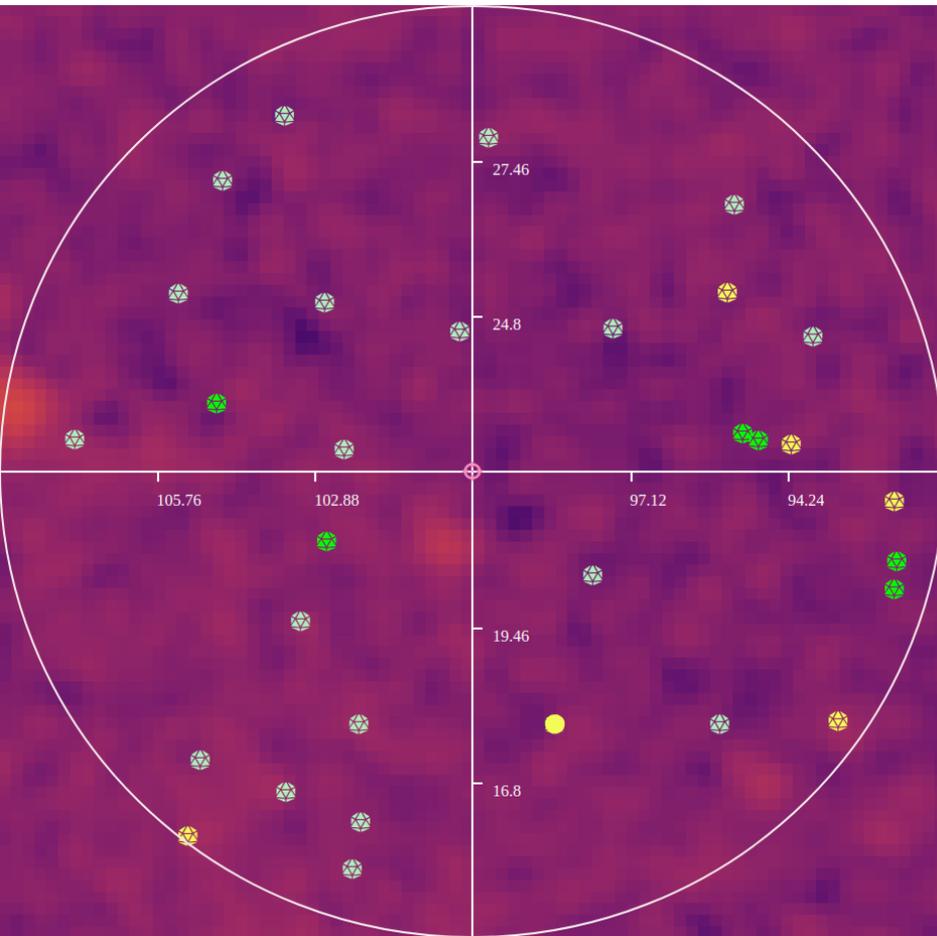
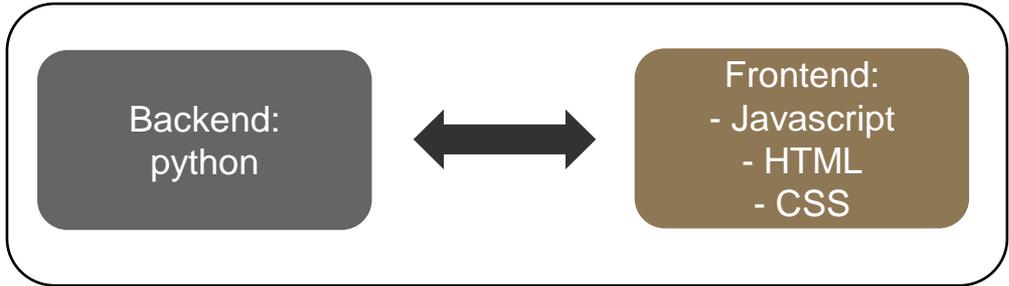
Architecture using
Eel framework

Backend:
python



Frontend:
- JavaScript
- HTML
- CSS

History of Colibri – v1



Status

Source name not known. Please remove text or choose known source!
 Internet connection: established
 Visibility plots: finished next night / finished this month
 Latest grbs: finished search
 Latest neutrinos: finished search

Search area

Source name: abc

RA*: 100 Dec*: 22.13 Radius*: 8

Time window

date & time*: 08/28/2020, 12:11:49 PM window [days]*: 50

VoEvent

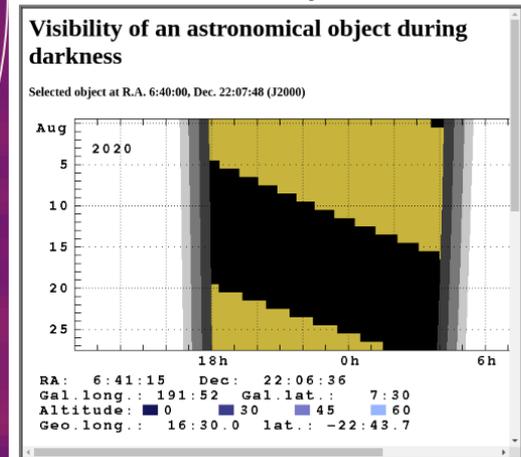
VoEvent: e.g. ivo://nasa.gsfc.gcn/SWIFT#BAT_GRB_Pos_848890-834

Catalogs

VoEventDB 4FGL TeVcat FLAapLUC

Latest transients

GRBs Neutrinos None

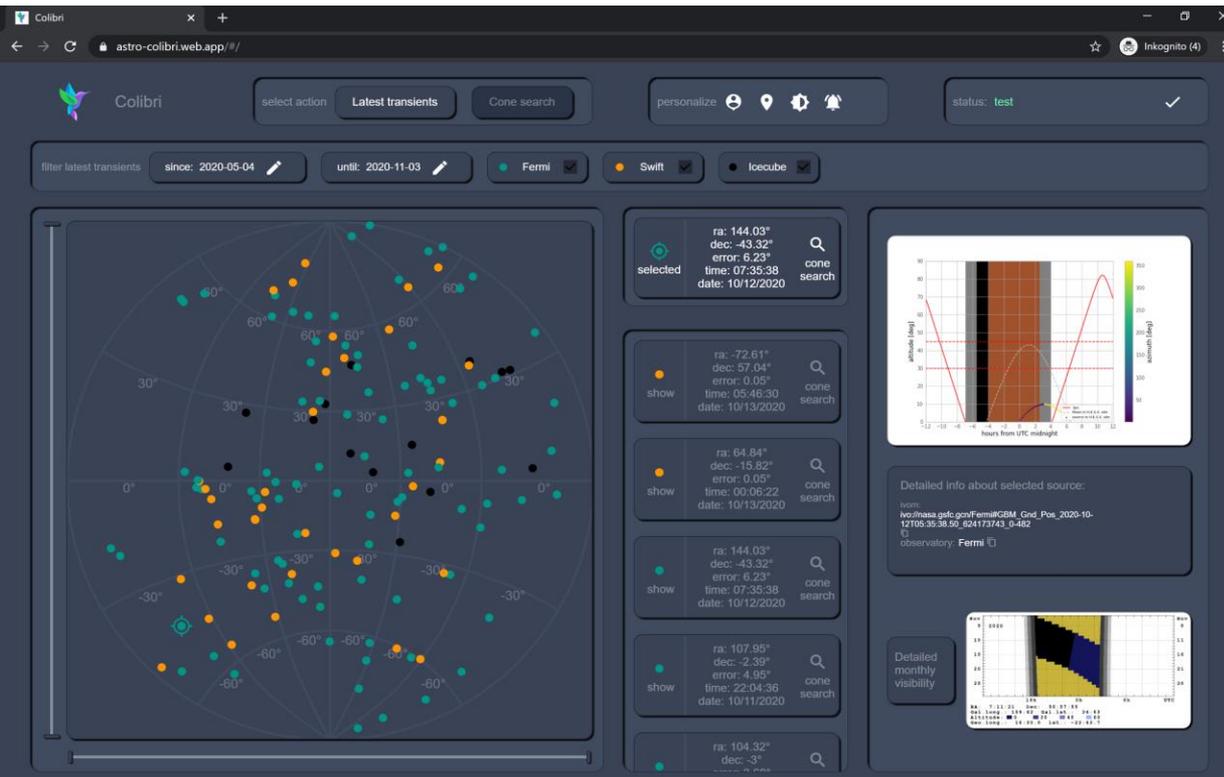
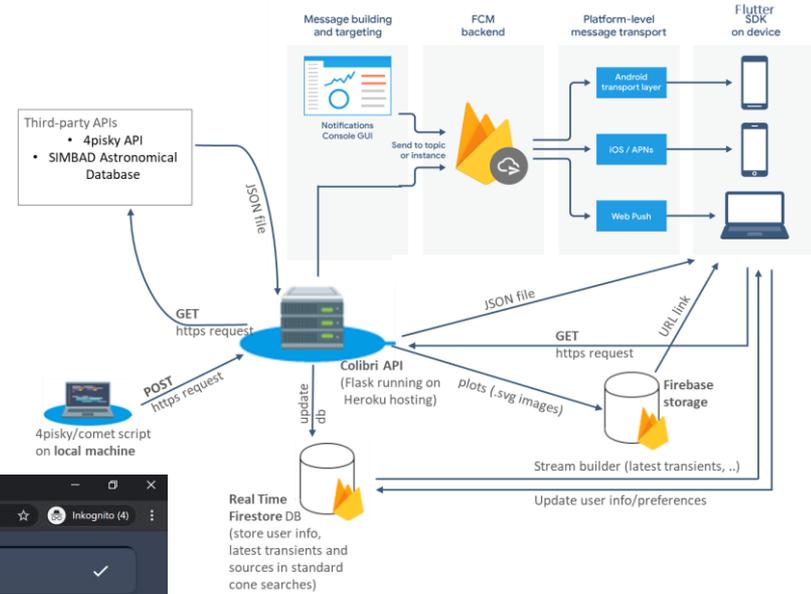


START



End of August 2020

History of Colibri – v2

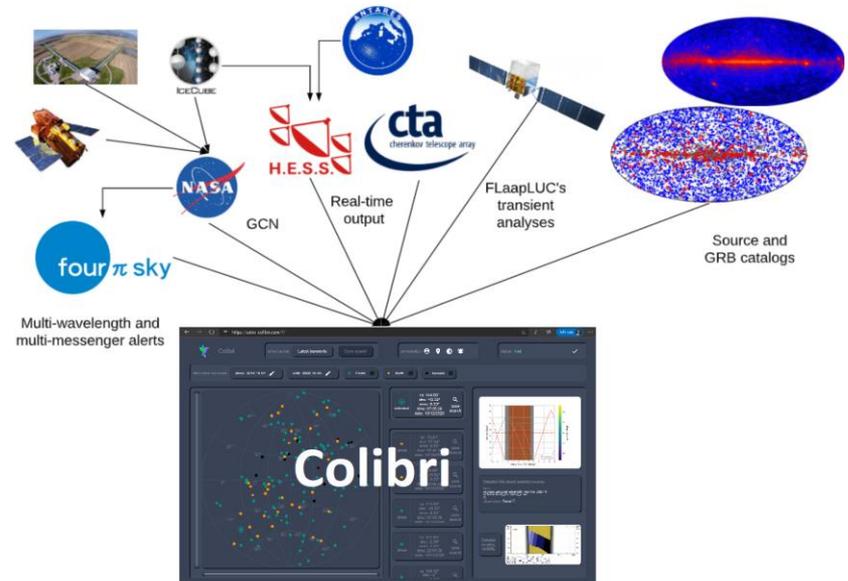


Section

Architecture

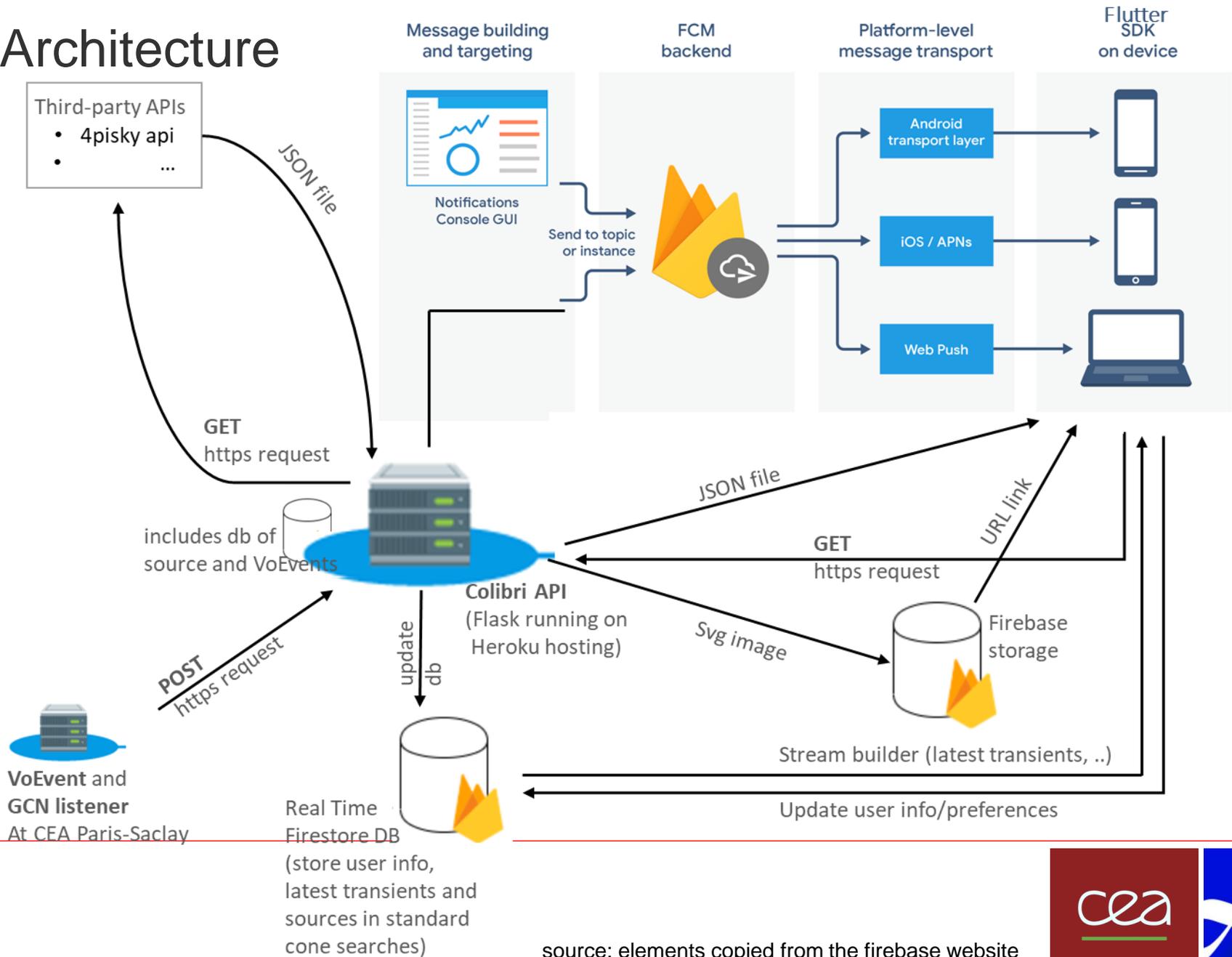
Building on existing systems

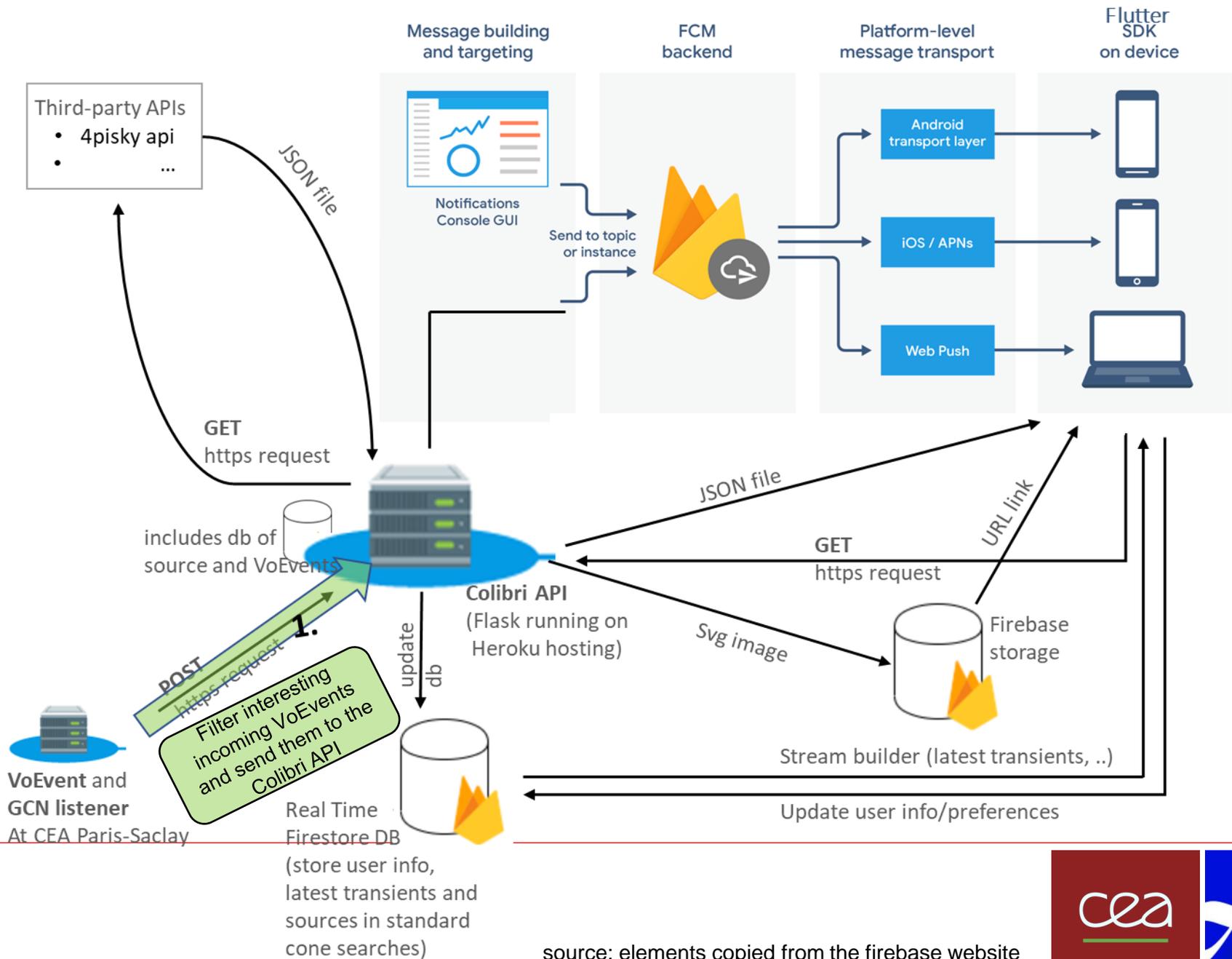
- Astro COLIBRI will use existing alert systems and catalogs (incomplete list):
 - VoEvent alerts (via 4pisky)
 - AMON alerts
 - GCN circulars
 - Fermi-LAT (4FGL + FlaapLUC + Fava)
 - TevCat catalog
 - Vera Rubin Observatory (e.g. FINK broker)
 - Vovisobs protocols for schedules/observations

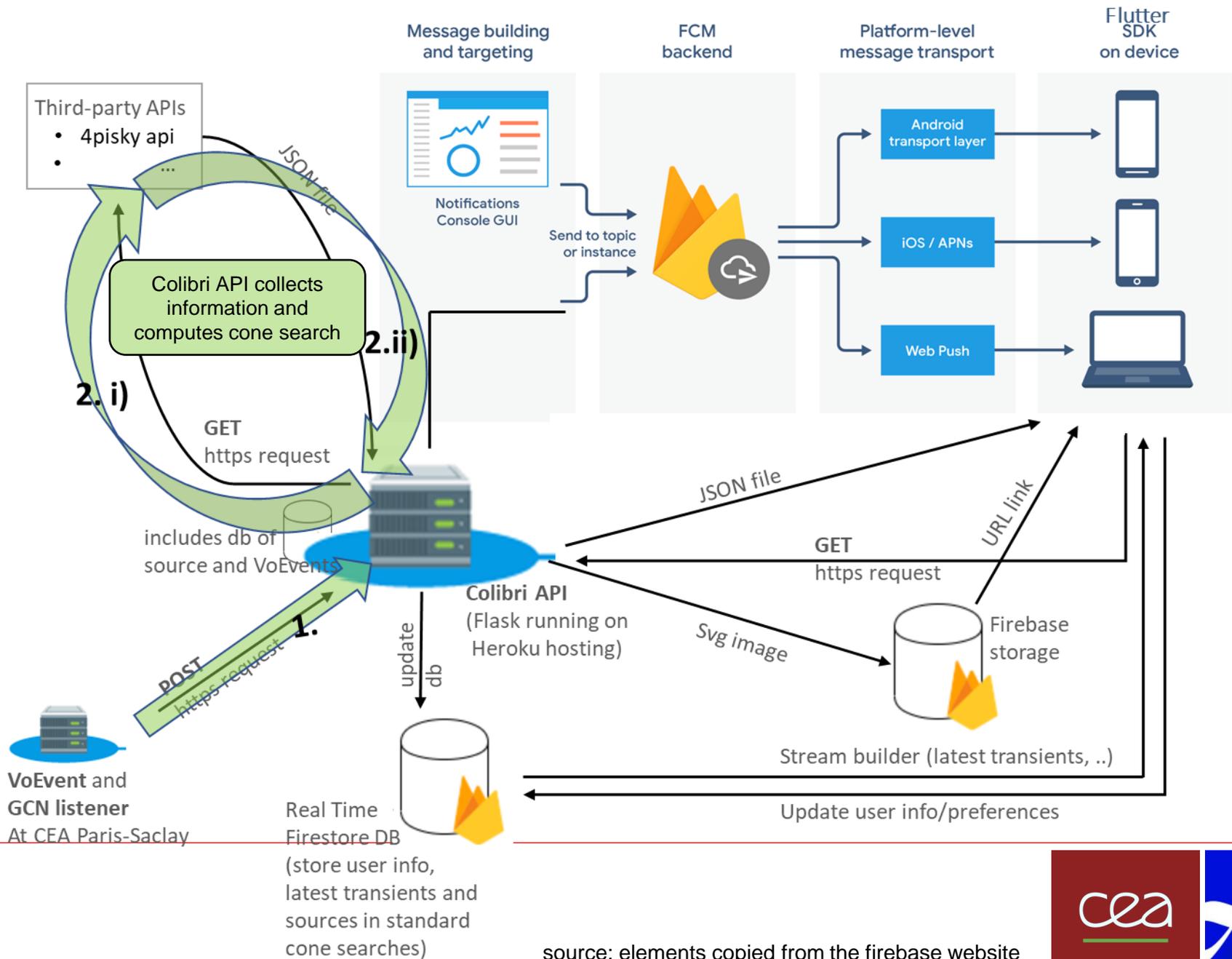


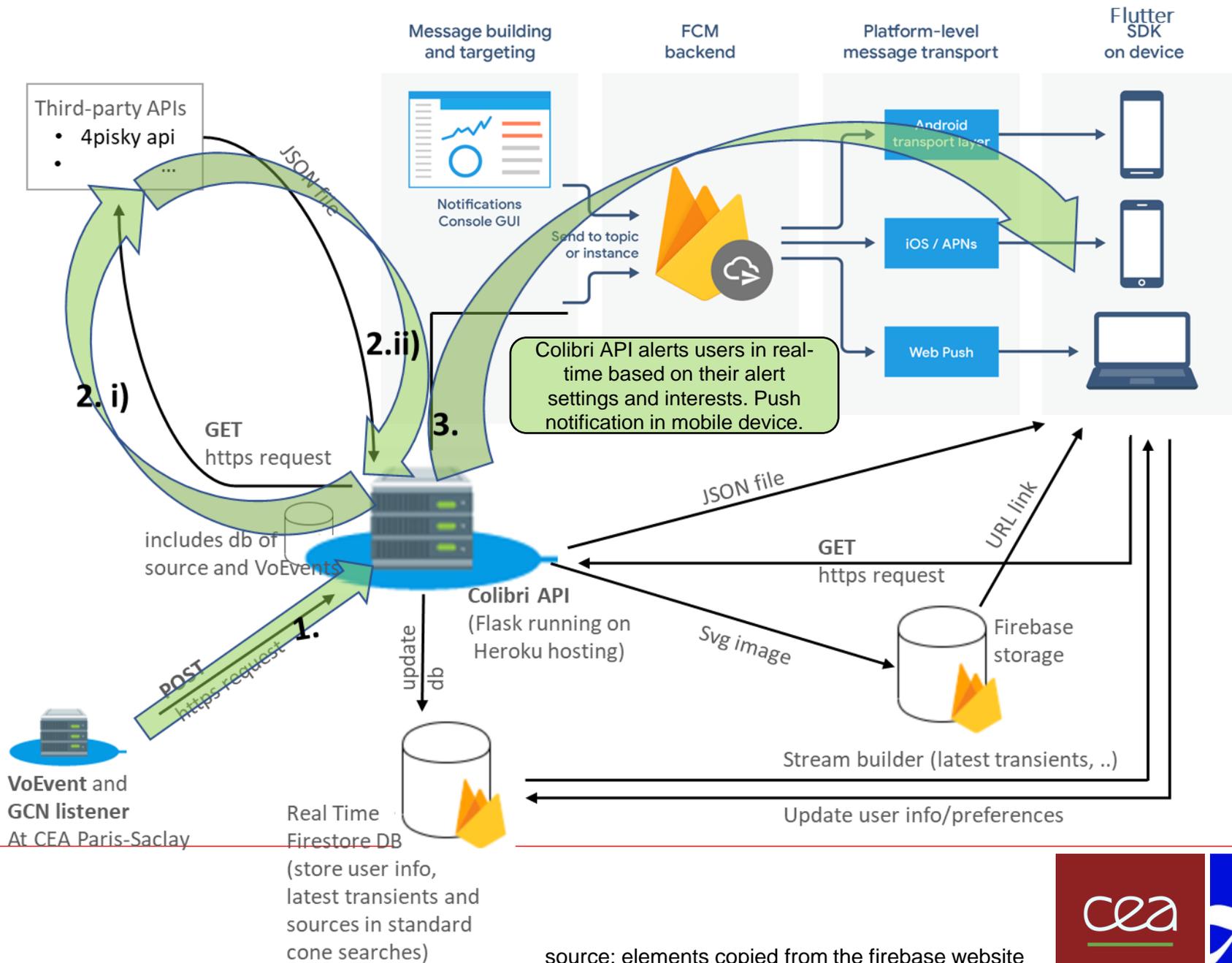
- Colibri aims to be the top layer that combines existing subsystems to one large ecosystem
- Dedicated also to humans through an intuitive GUI
 - Makes communication about possible observations easy
 - In some cases the observation committee needs to be convinced

Architecture





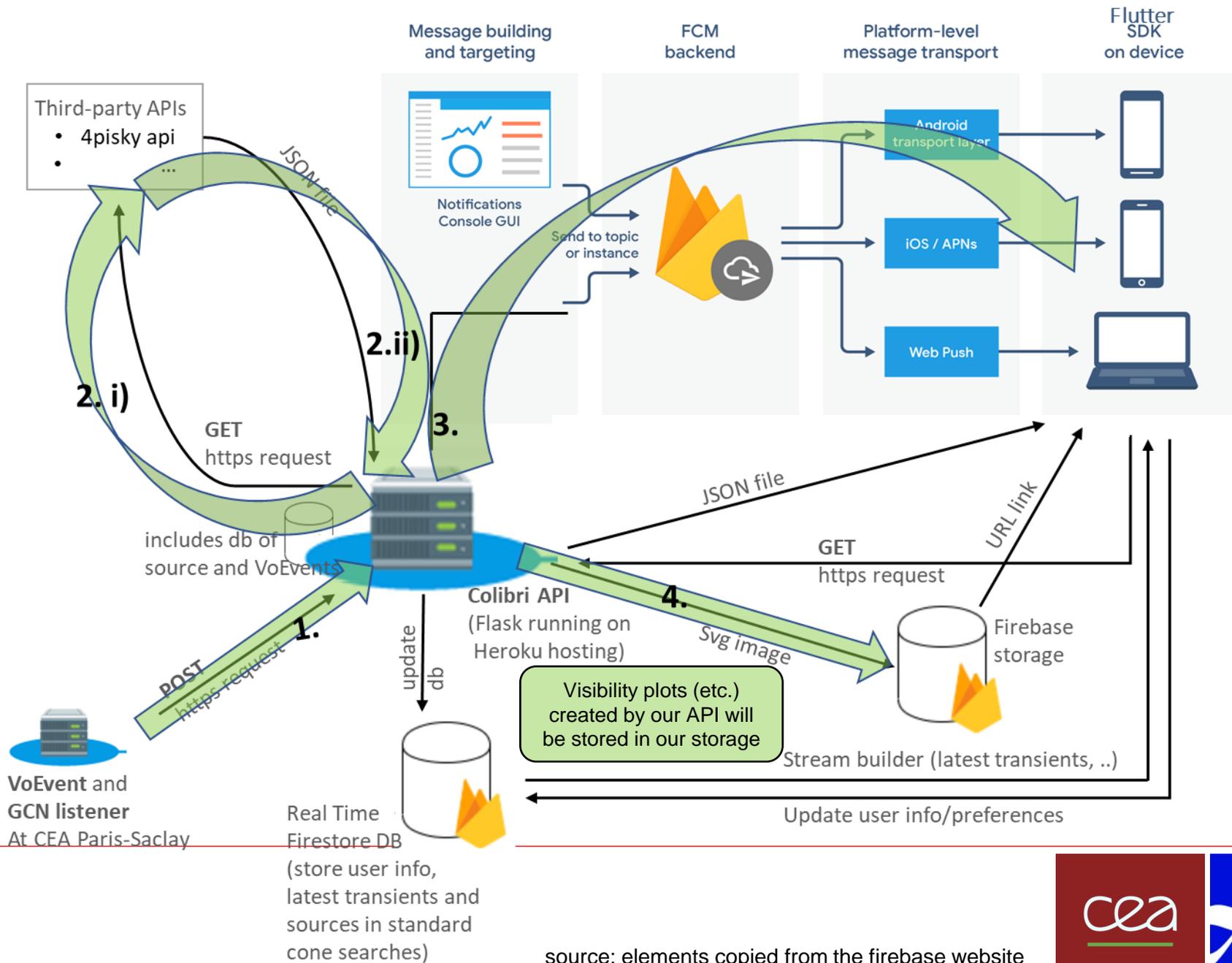


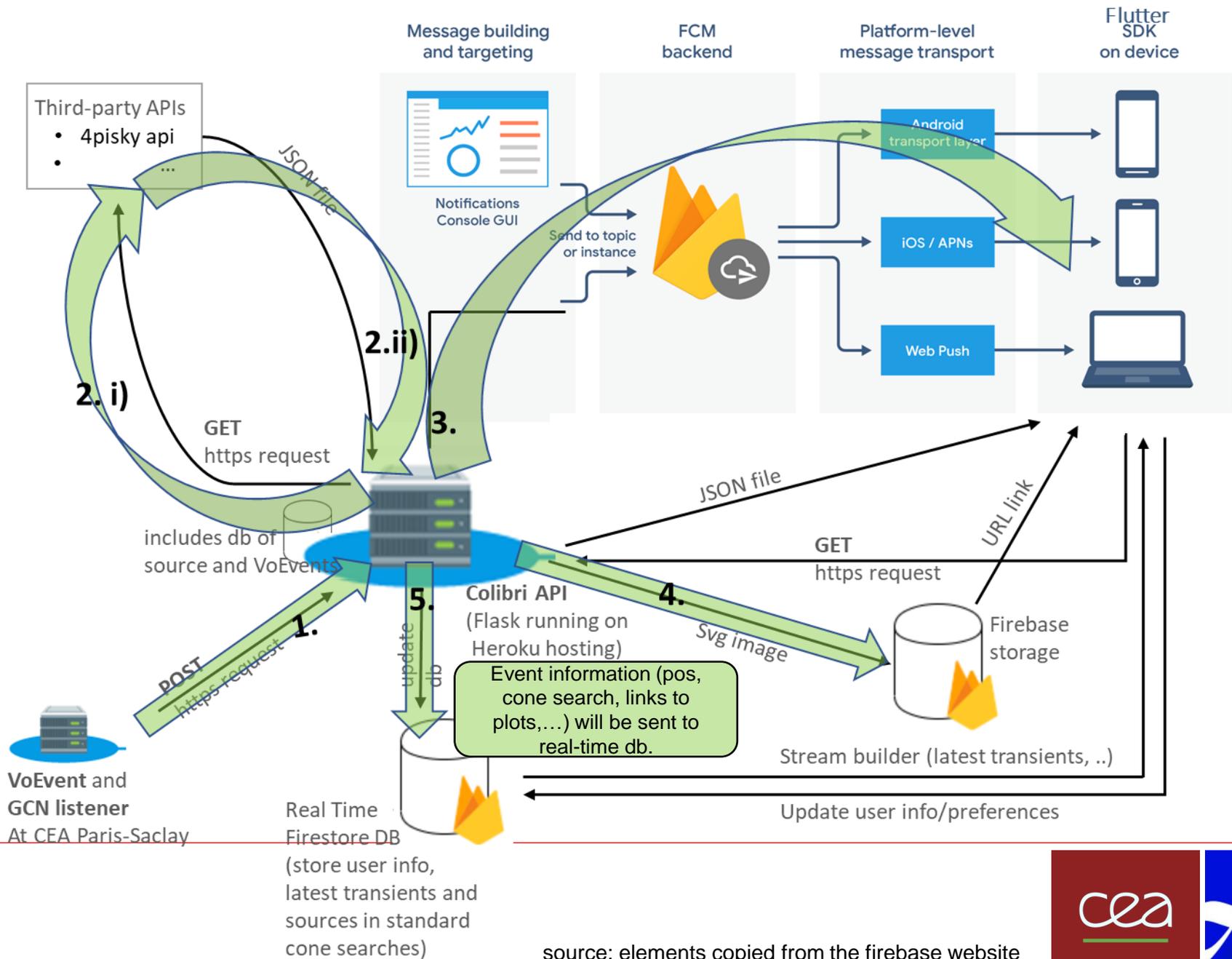


Architecture – Astro-COLIBRI Clients

- Same code base for website and app using Flutter framework
 - Dart programming language
 - Code will be converted to JavaScript, HTML and CSS for the website
 - Apps will be published in *Google Play* store and *iOS App* store
- Hosting (astro-colibri.com)
 - Compressed files are cached on SSDs at Content Delivery Network (CDN) edges around the world
 - Fast access to our website from everywhere



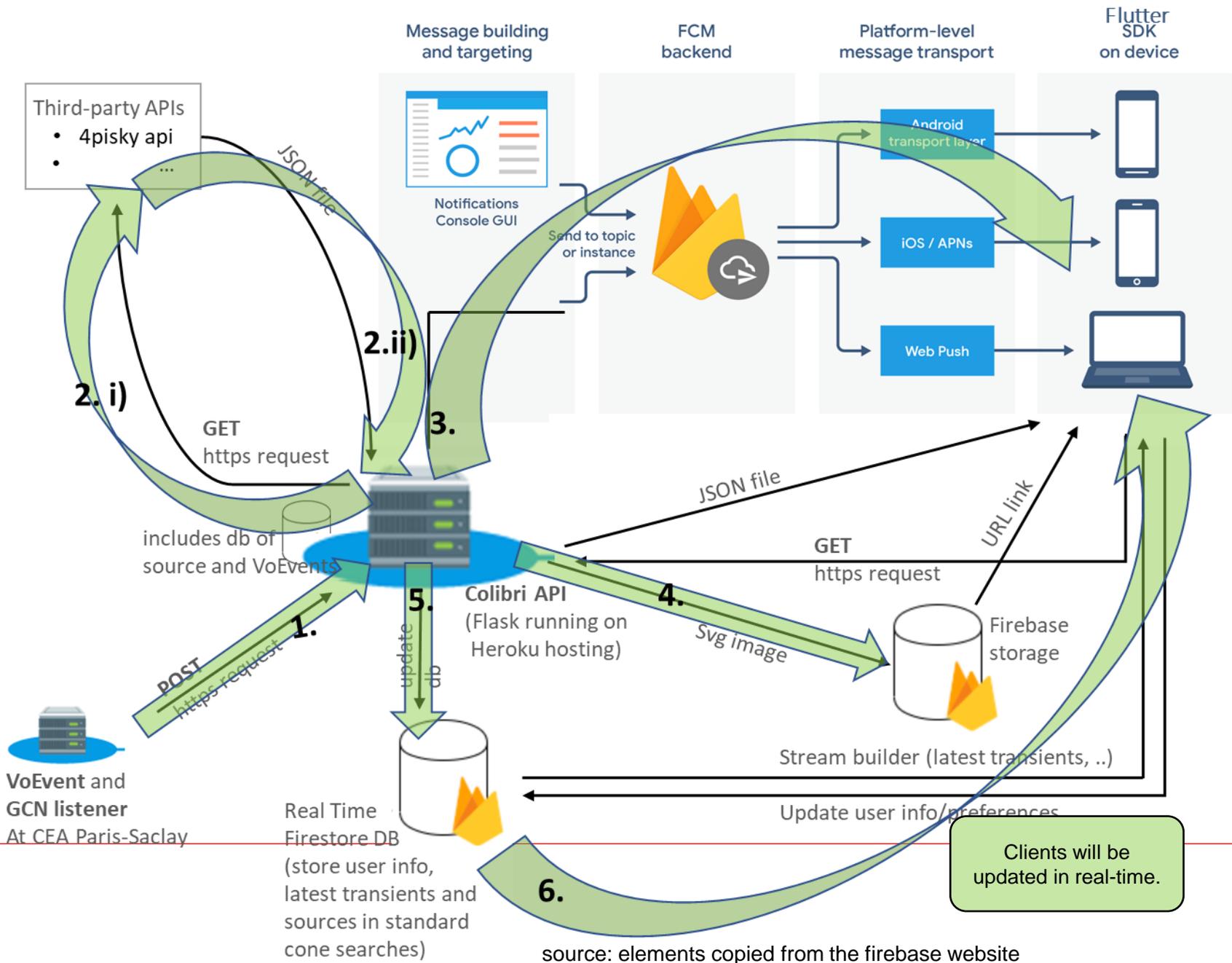


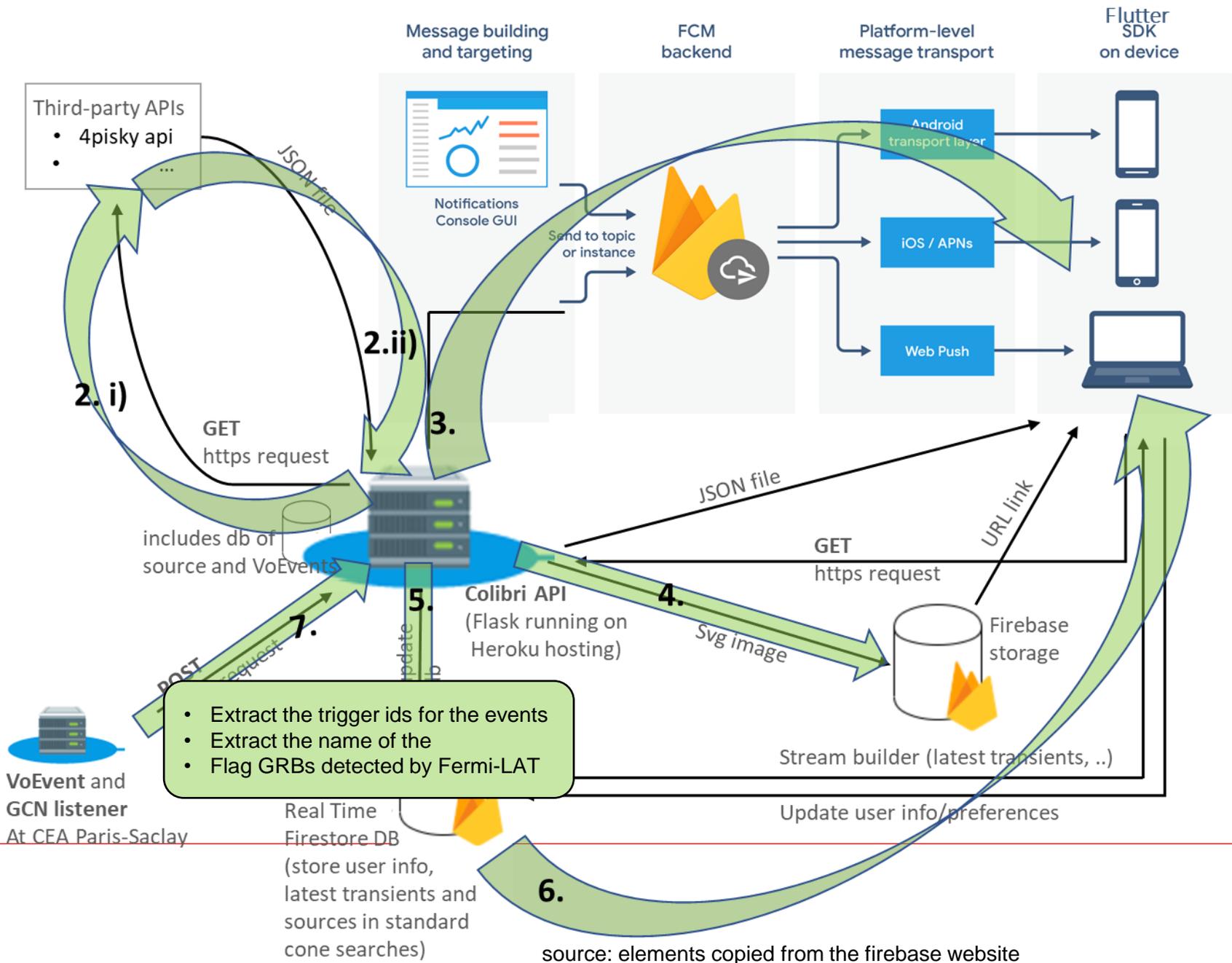


Architecture – real-time database (DB)

- Offline support → **better user experience**
 - DB caches data that Colibri app is using often
 - When device comes back online, all changes will be synchronized back to DB
- Expressive querying → **better performance**
 - Chained and combined filters and queries supported
 - Multi-index database queries
 - Efficiently merges indexes using a zig-zag merge join algorithm
- Flexibility → **easy extension**
 - Data is stored in documents, organized in collections
 - Documents can contain complex nested objects in addition to subcollections
- Realtime → **better performance**
 - Usage of data synchronization via streams
 - Every time data change, any connected device receives that update within milliseconds







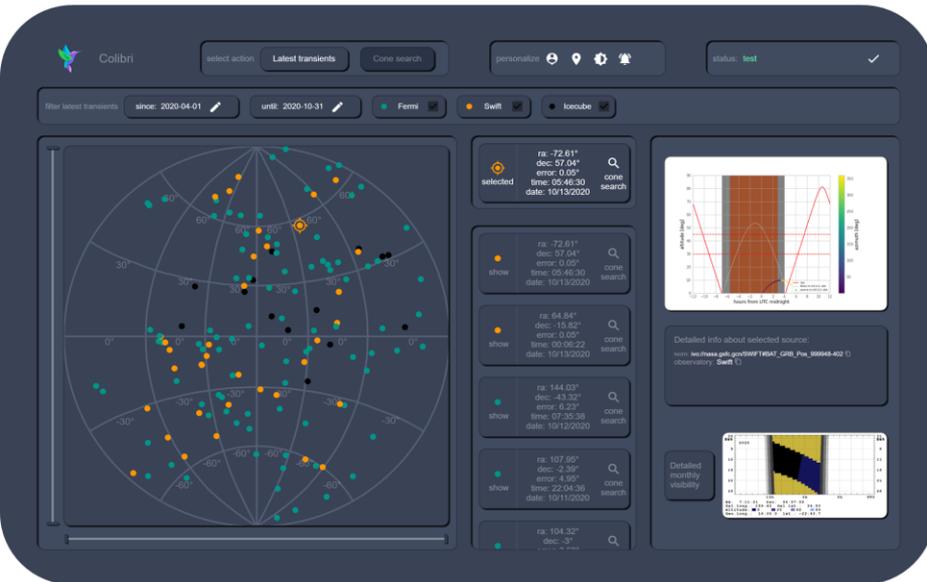
Section

Current use cases in beta version

Current use cases

Latest transients

- ❖ **Python script** runs on our local server that listens to incoming VOEvent messages of astronomical observations
- ❖ Interesting events will be passed to the **Colibri API**
 - further data collection
 - correlation with archival data (cone search around signal, combine with previous alerts, ..)
 - calculating visibility plots for observatories
 - alerting clients based on user settings, preferences, and visibility
- ❖ **Clients** are updated via a stream builder connection to our **real-time database**
 - clients show latest transients in map
 - latest transients are also displayed as cards ordered by time



Cone search

Input:

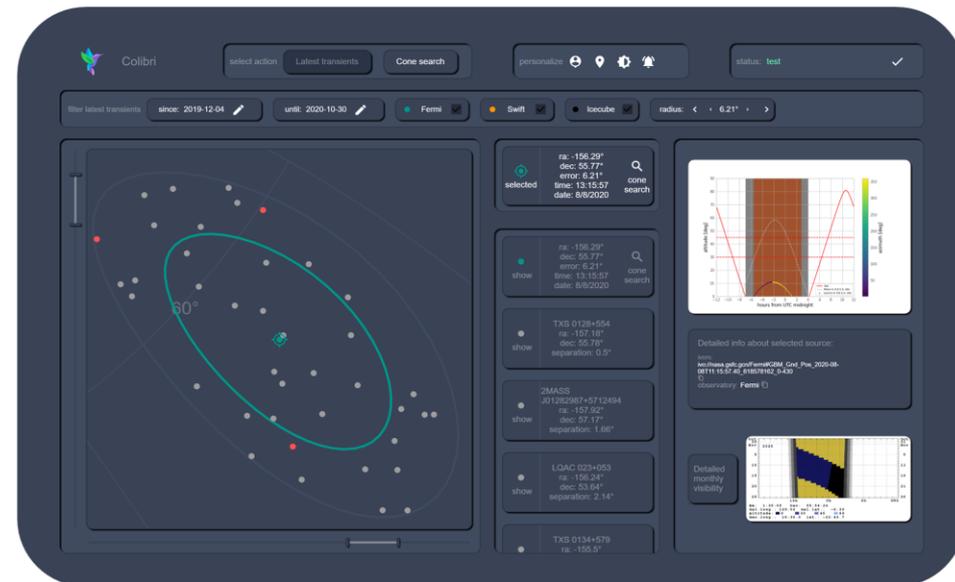
- ❖ Latest transient (click cone search button in latest transient overview)
- ❖ User specified input coordinates (coming soon)

Computation:

- ❖ Cone search of a latest transient is loaded from real-time db
 - fast display of cone search
- ❖ User specified input coordinates require a new cone search via the Colibri API
- ❖ API sends computed data back to clients

Display:

- ❖ Clients display latest transient in center with boundary that corresponds to signal uncertainty
- ❖ Sources are shown in zoomable map
- ❖ Clicking on sources provides further information in the right column (visibility, links to other websites, ...)
- ❖ Sources displayed in overview cards that are ordered by separation from cone search center



Section

Outlook

Outlook

- Finalizing the beta version within next months
- Including gravitational wave alerts
- Distribute mobile app via Google Play store and iOS App store
- Distribute among burst advocates in observatories
- Links or implementation within the real-time analyses' frameworks of current (and future) observatories

