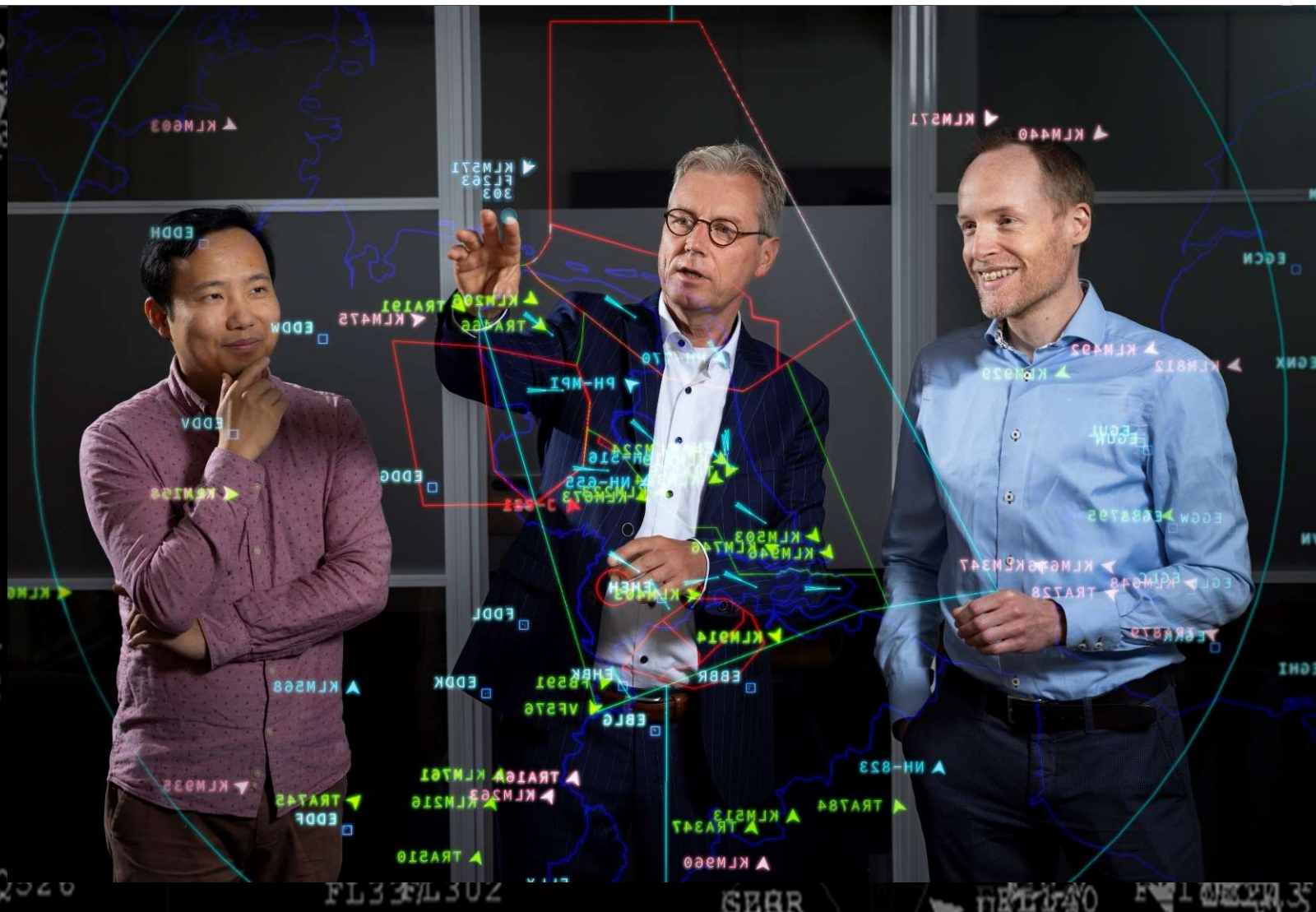


A Radically Open Source, Open Data, No License Approach to Air Traffic Simulation

Jacco Hoekstra, Joost Ellerbroek
Junzi Sun (Open AP part)



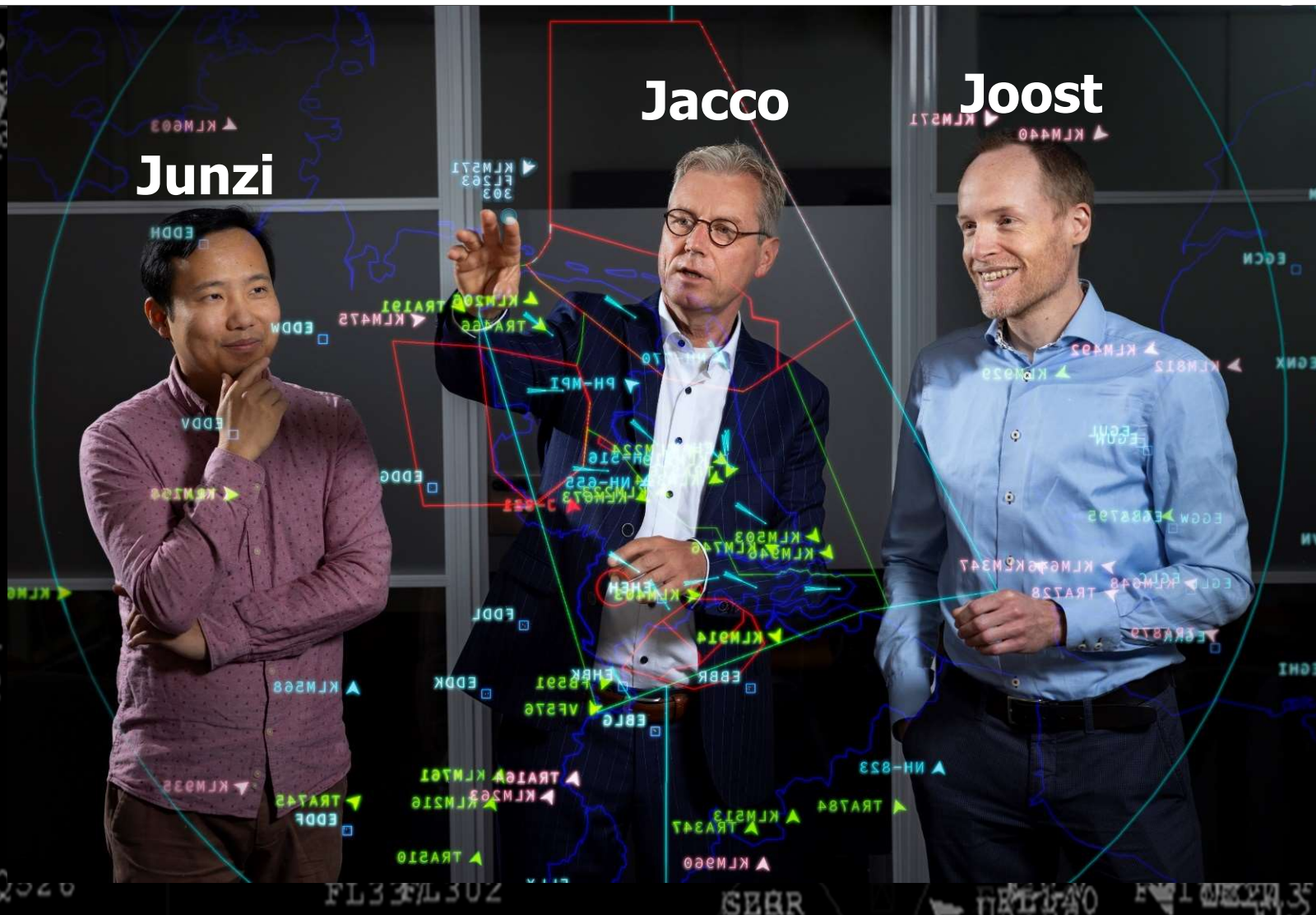
A Radically Open Source, Open Data, No License Approach to Air Traffic Simulation

Jacco Hoekstra, Joost Ellerbroek
Junzi Sun (Open AP part)

Junzi

Jacco

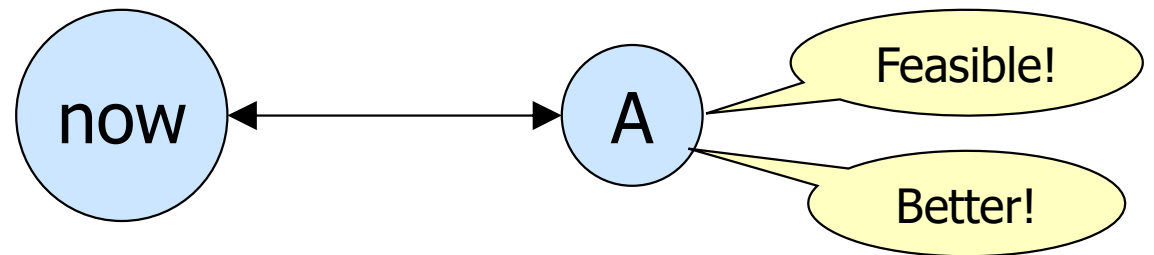
Joost

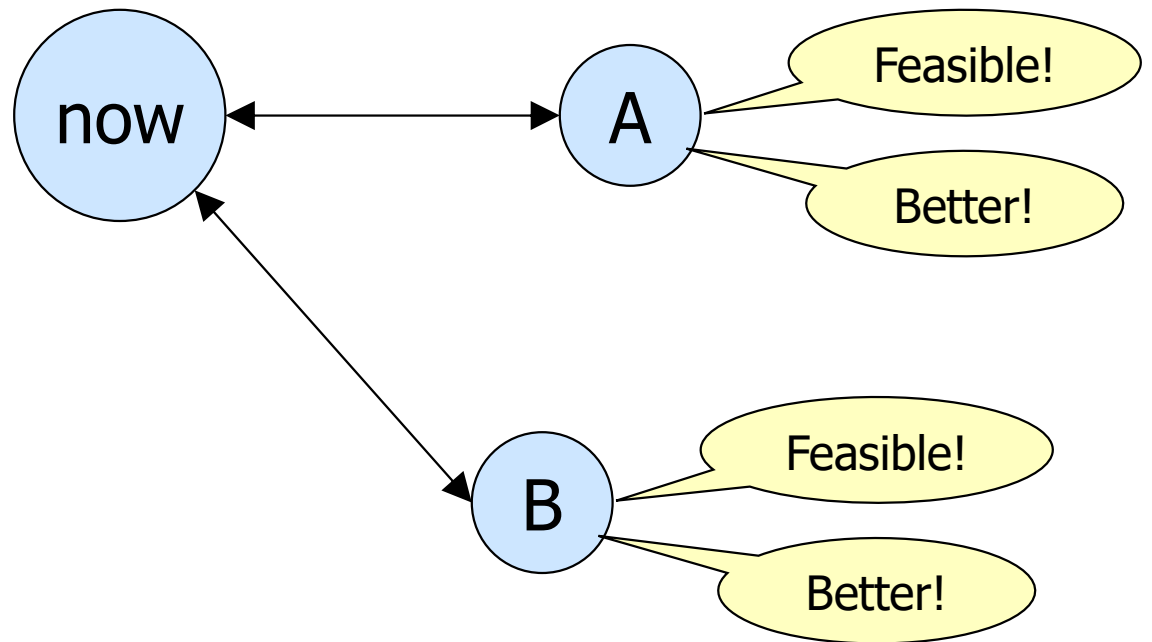


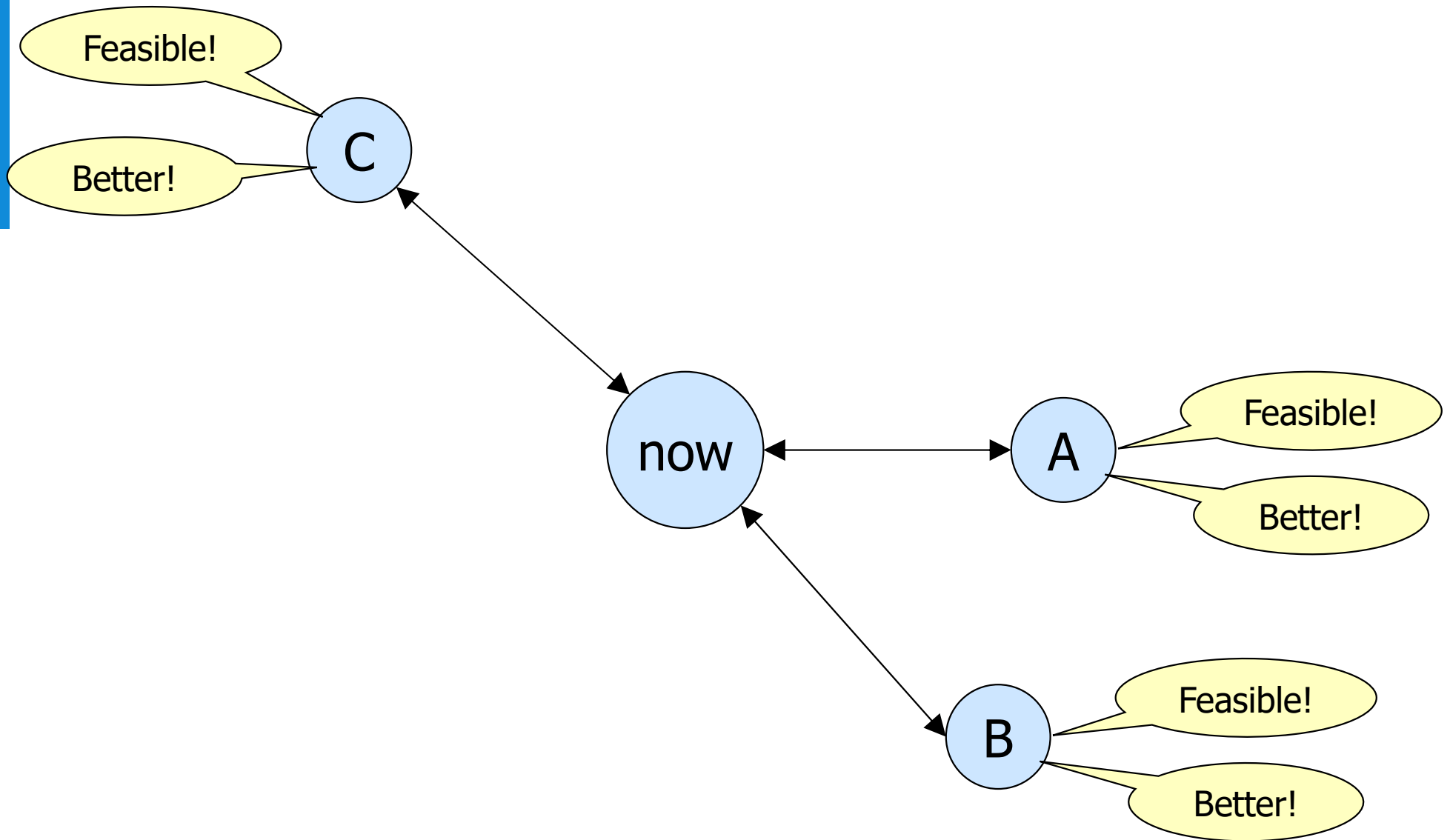
A Radically Open Source, Open Data, No License Approach to Air Traffic Simulation

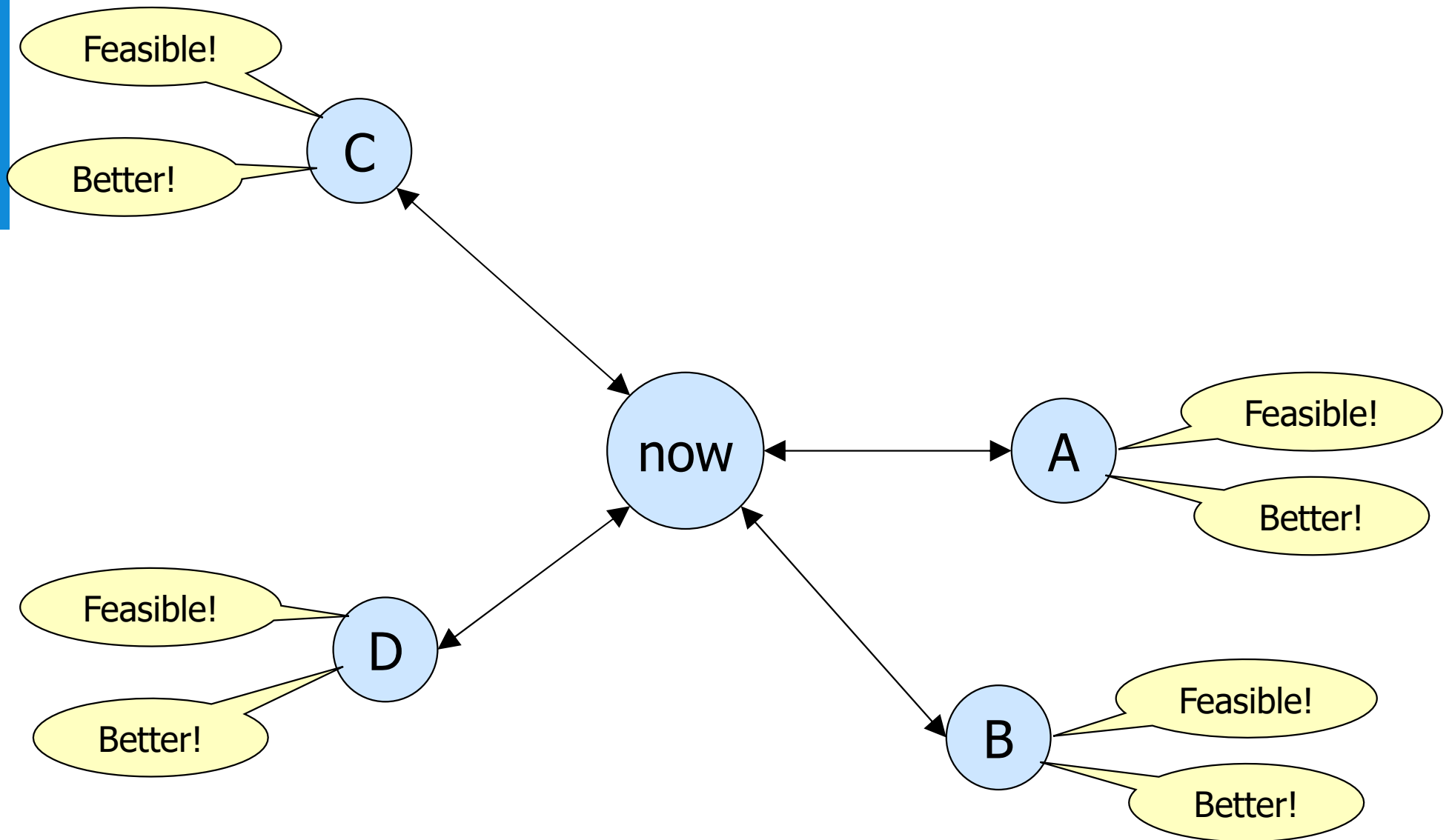
Jacco Hoekstra, Joost Ellerbroek
Junzi Sun (Open AP part)

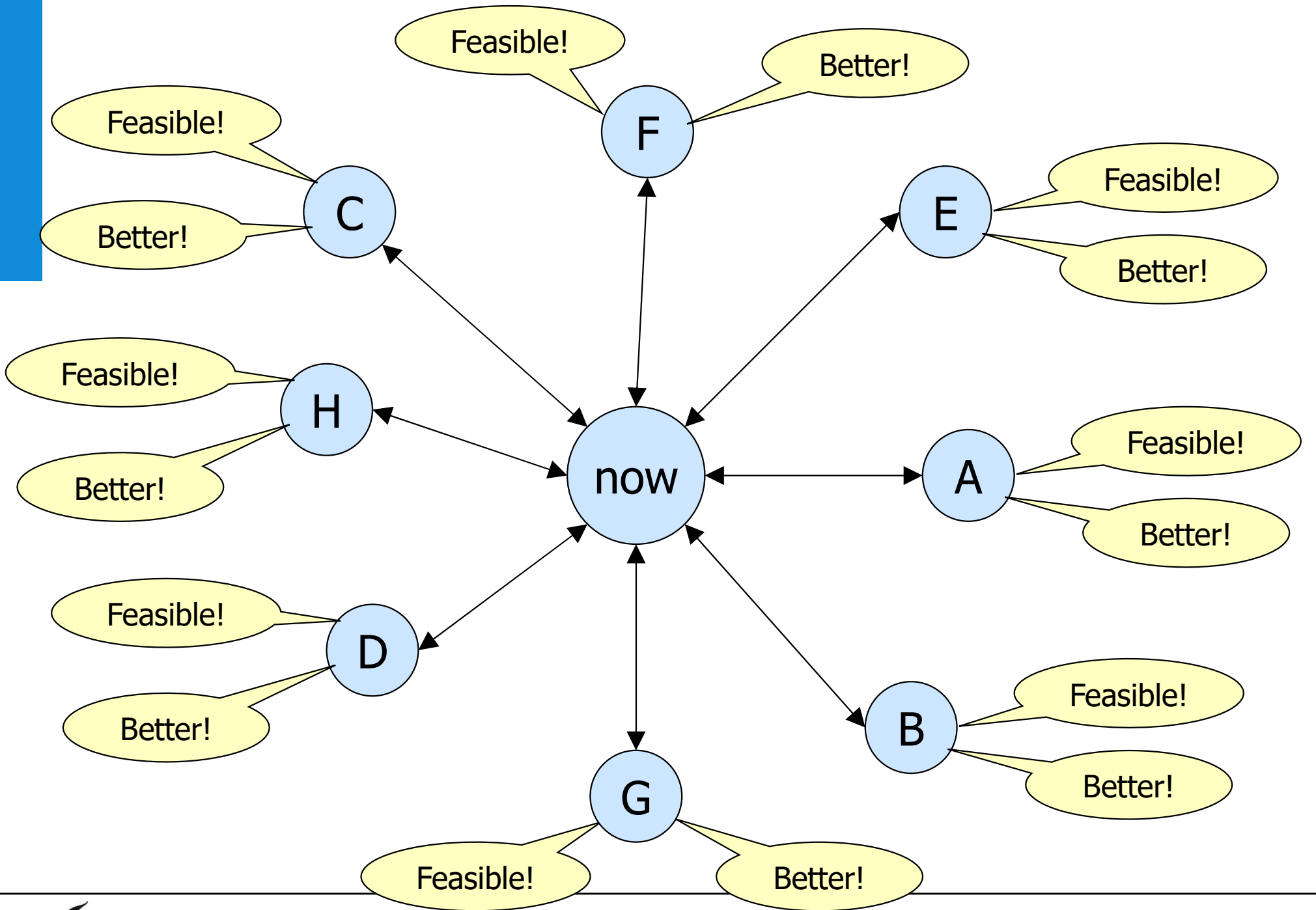
Current status of ATM Science











Research lines CNS/ATM group TU Delft

1. Foundation of ATM research

Google: TU Delft *ADS-B Junzi Sun*

1. Open source tools (ADS-B decoding, ATM simulation, conversion tools)
2. Open Data (Performance data, scenarios, historic traffic data)
3. Big Data & Metrics (Complexity, Traffic Flow dynamics)

Google: TU Delft *BlueSky*

2. ADS-B surveillance and applications

Google: TU Delft *OpenAP*

1. Raw data analysis, surveillance quality
2. Airborne Applications (ATSAW, CD&R algorithms)

3. Drone CNS/ATM technology

1. Indoor & outdoor navigation and autonomy
2. Swarming

4. General Aviation CNS/ATM technology

1. Airborne radar & surveillance technology for safety

5. Schiphol/Mainport related CNS/ATM issues

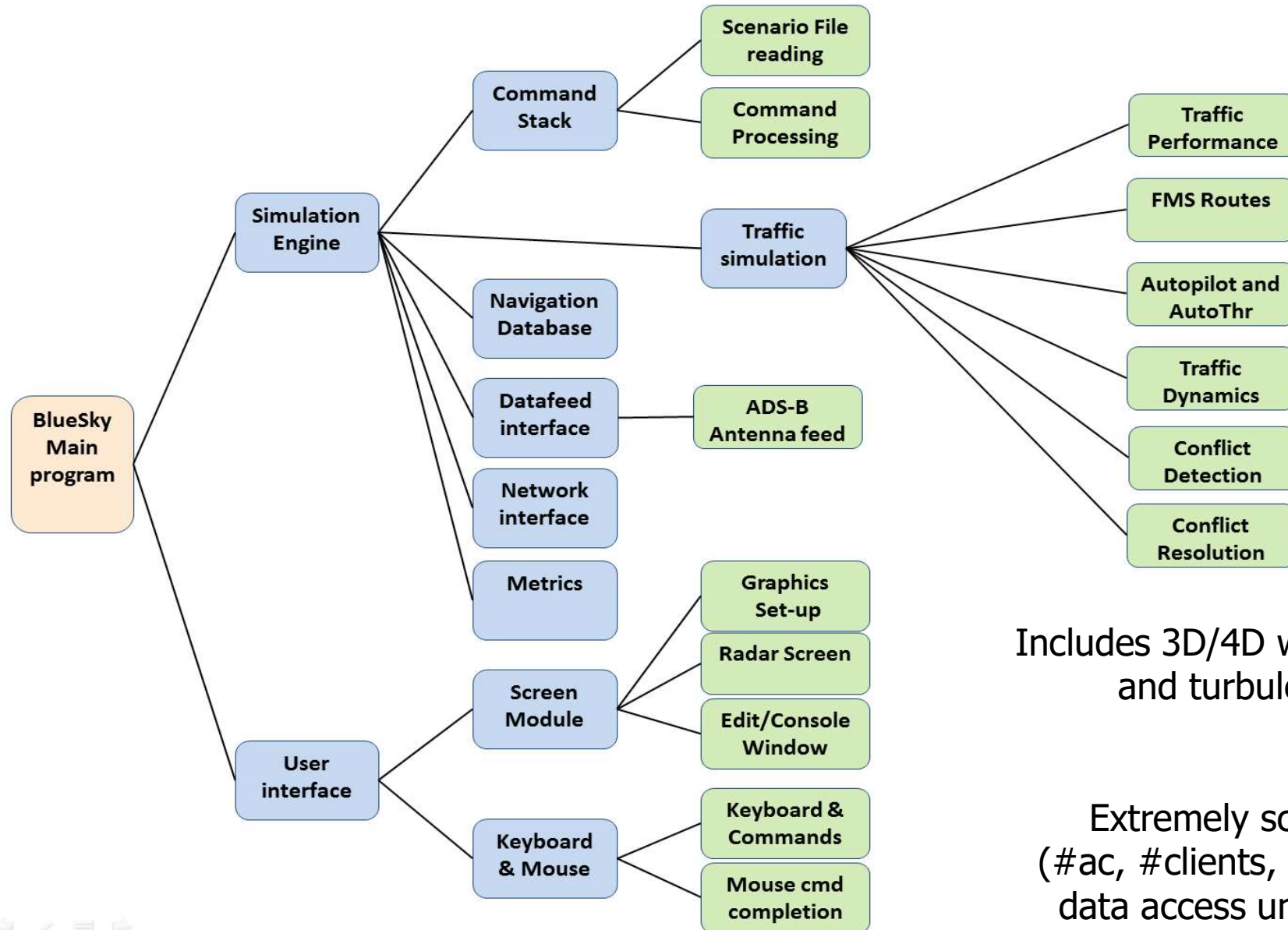
Improve science & research in ATM

- **Comparable** research results, same metrics, tools and scenarios
- Research does not thrive under **standardisation**
- But every **researcher needs** aircraft performance models => success
BADA 3
- **Open Source, free** (also for tools)
- **Open Data, free**

Inspiration: Wikipedia & BADA

- Data files with performance and procedure data
- BADA 3 vs BADA 4
- Quality BADA 3 could be improved easily
- BADA 4 has more sensitive, proprietary data
- BADA 3 currently default standard for research
- Licensing regime tightens
- Not fully open (sharing/publishing not allowed)

Modules inside BlueSky Air Traffic Sim



Includes 3D/4D wind model and turbulence

Extremely scalable
(#ac, #clients, #plug ins
data access unlimited)

BlueSky Air Traffic Simulator

- 100 % fully open data:
 - Aircraft performance data
 - Navigation data
- 100 % open source:
 - Co-development shared via [GitHub/TUdelft-CNS-ATM/BlueSky](https://github.com/TUdelft-CNS-ATM/BlueSky)
 - Should not require any commercial development tools or libraries
 - Quality control?
- Increase adoption:
 - Very fast: simulate 10,000+ vehicles fast-time incl FMS, autoliot, performance etc.
 - Allow plugins to extend functionality without need to understand rest of the code
 - Easy to use: GUI and data/scenarios in plain text files, run locally, Readable source code for scientists, who are not computer experts
 - Philosophy: no need to understand what you do not need to know

Open Source: Python language of choice

- BlueSky simulator made entirely in Python
- Based on standard scientific libraries
- Python 3.10+
- Runs on Windows, Linux & MacOS
- Uses Numpy, Scipy
- For GUIs:
 - Qt, OpenGL,zmq
 - baseline: SDL (via Pygame)
- See requirements.txt or setup-python.bat for all dependencies



BlueSky GNU General Public License v3

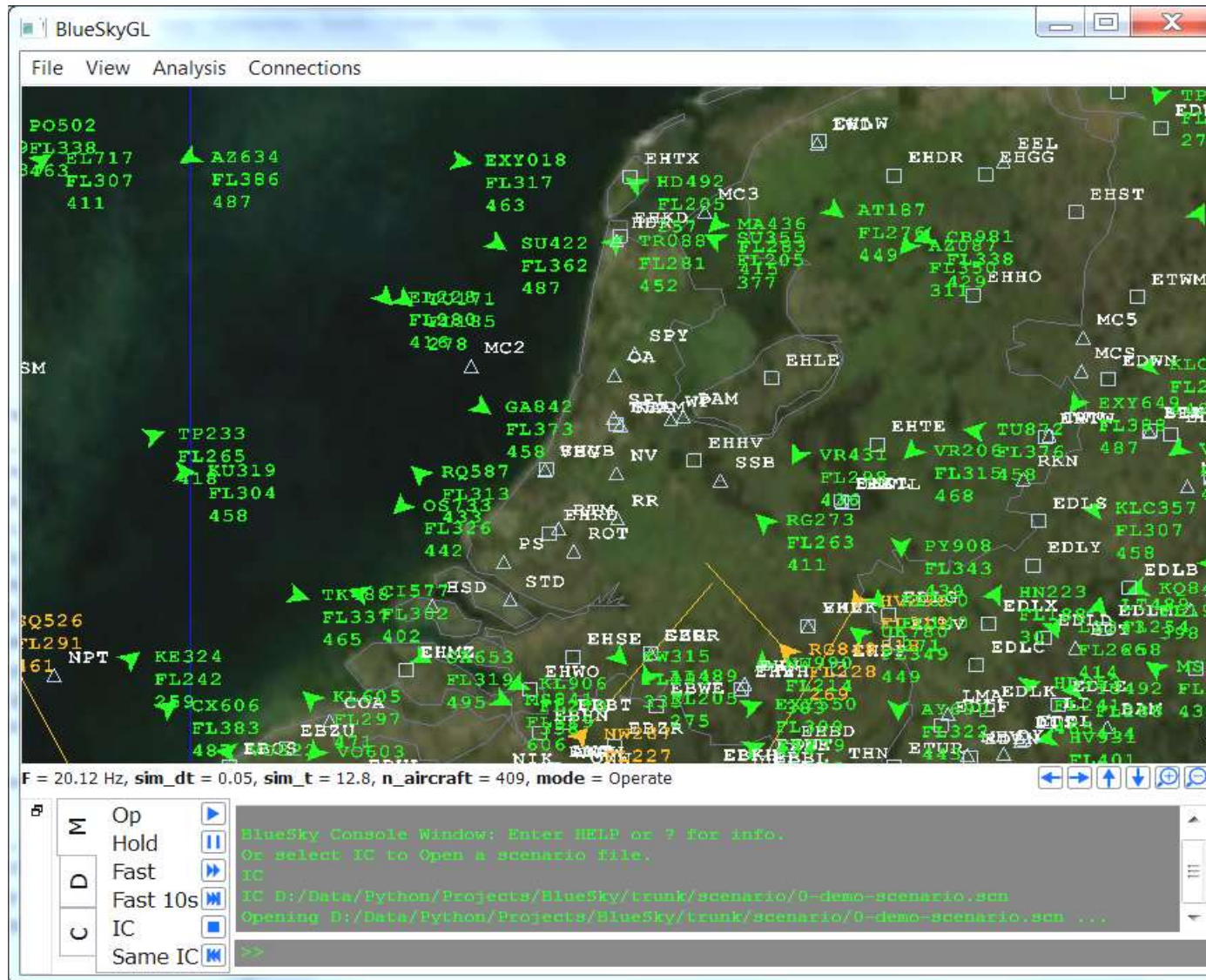
- **Required:**
 - Disclose Source
 - License and copyright notice
 - State Changes
- **Permitted:**
 - Commercial Use
 - Distribution
 - Modification
 - Patent Use, Private Use
- **Forbidden:**
 - Hold Liable



"Funny, really. They always click on 'install' without reading the license agreement."

- <http://www.gnu.org/licenses/quick-guide-gplv3.en.html>

GUI: Standard (Qt/OpenGL based)



GitHub site contains documentation

The screenshot shows a web browser window displaying the GitHub Wiki page for the repository 'TUDelft-CNS-ATM/bluesky'. The page title is 'Home' and it was last edited 5 days ago. The main heading is 'BlueSky - The Open Air Traffic Simulator'. The page content includes a description of BlueSky as a research tool for Air Traffic Management and Air Traffic Flows, and a 'Table of Contents' section with a list of links to various documentation topics. A sidebar on the right shows a list of 148 pages, including 'Home', 'addnodes', 'addwpt', 'ADDWPTMODE', 'aero', 'AFTER', 'Aircraft ID', 'Aircraft settings', 'alt', and 'Altitude'.

Home
Joost Ellerbroek edited this page 5 days ago · 28 revisions

BlueSky - The Open Air Traffic Simulator

BlueSky is meant as a tool to perform research on Air Traffic Management and Air Traffic Flows. Its goal is to provide everybody who wants to visualize, analyze or simulate air traffic with a tool to do so without any restrictions, licenses or limitations. It can be copied, modified, cited, etc. without any limitations.

This wiki provides the documentation for installing, running, and modifying BlueSky.

Table of Contents

- [Installation](#)
- [Tutorials](#)
- [Running BlueSky](#)
 - [Command line options](#)
 - [Basic Operation](#)
 - [Data Logging](#)
 - [Batch Simulation](#)
 - [Network Communication](#)
 - [Command Reference](#)
 - [Troubleshooting](#)
- [Programming BlueSky](#)
 - [Optimization through Vectorized Operations](#)

Pages 148

Find a Page...

- ▼ [Home](#)
 - BlueSky - The Open Air Traffic Simulator
 - Table of Contents
- ▶ [addnodes](#)
- ▶ [addwpt](#)
- ▶ [ADDWPTMODE](#)
- ▶ [aero](#)
- ▶ [AFTER](#)
- ▶ [Aircraft ID](#)
- ▶ [Aircraft settings](#)
- ▶ [alt](#)
- ▶ [Altitude](#)

Plug-in: Easy way to code your extension

- `Init_plugin()` function:
 - Define the frequency with which your update should be called
 - Add your own user command to the dictionary, connect them to your own added functions. Uses built-in argument interpreter if you want.
- `Update()` function:
 - Called each time step (optionally before traffic update with `preupdate`)
- `Reset()` function:
 - Reset everything for next scenario run
- Input:
 - Access freely all variables e.g. `traf.gs[idx]` or functions e.g. `traf.id2idx("KL204")`
- Output:
 - `stack.stack('CRE KL642,KJFK,RW22L,0,0')`

Challenges Open Data vs Quality?

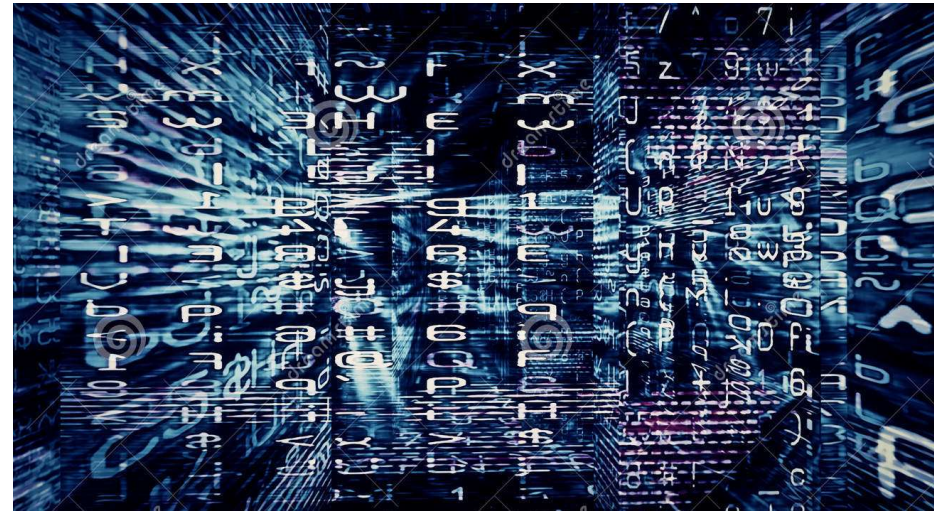
- **Navigation data:**

- Geographical information
- Navigational Aids
- Waypoints
- Airport data: taxiways, runways
- Sector lay-out/Airspaces

- **Aircraft Performance data:**

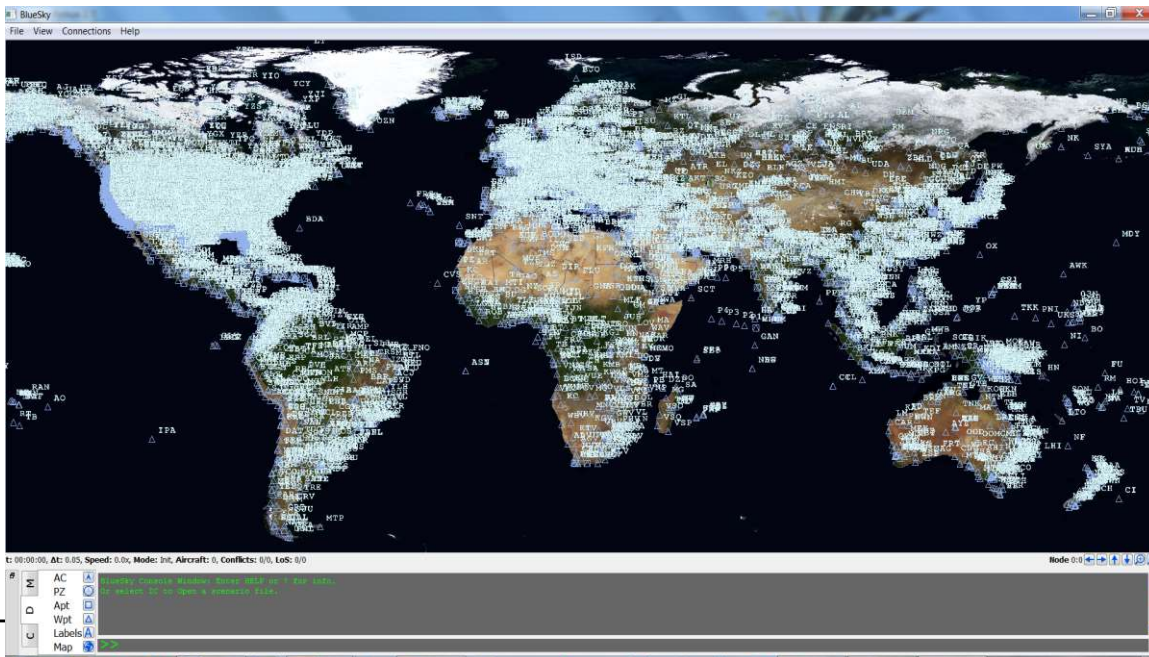
- Drag polar
- Engine performance
- Operating weights
- Autopilot/Autothrottle settings, mode logic
- Procedural speeds

- **Weather data?**



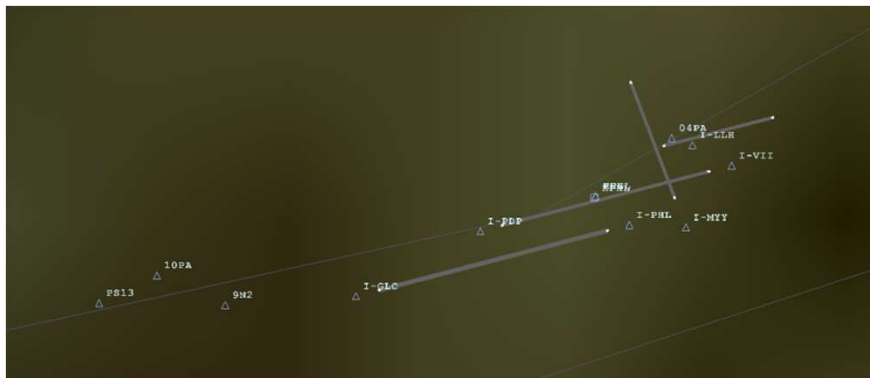
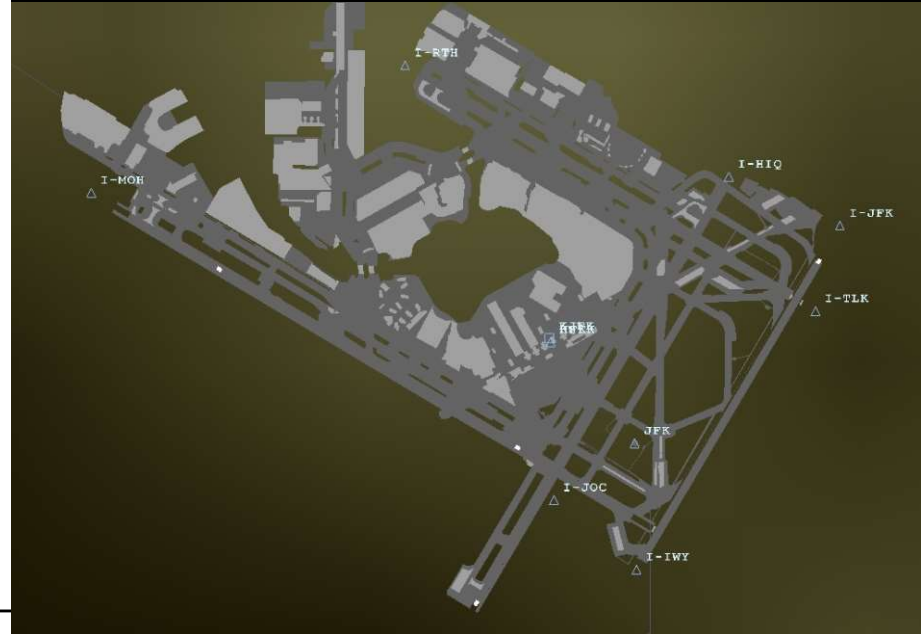
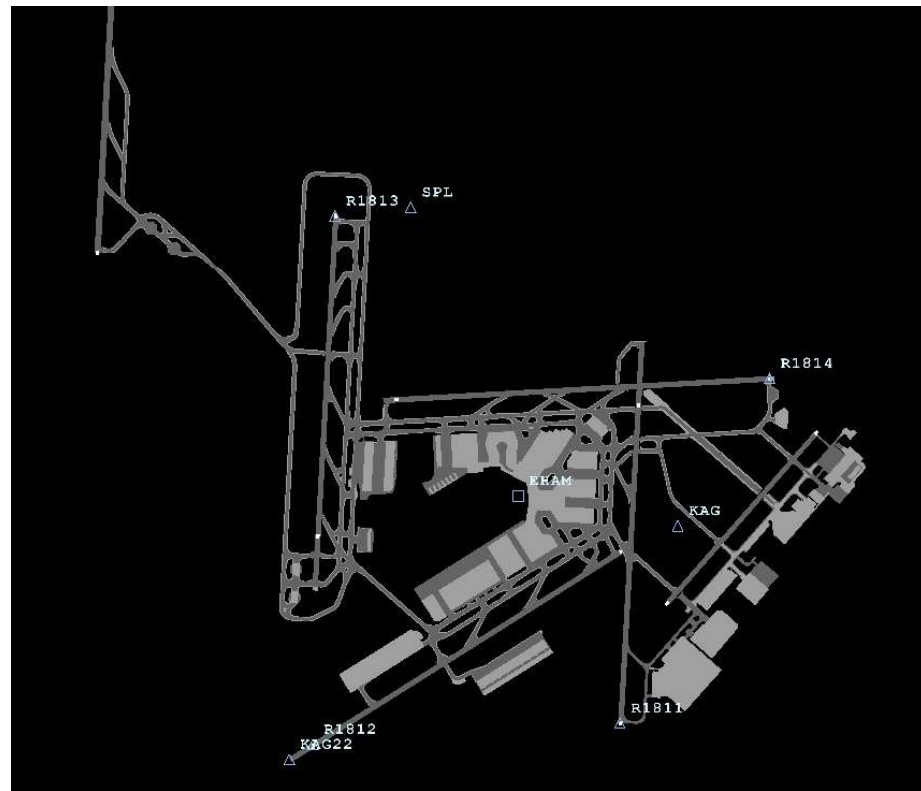
Open data: nav aids

- Web-crawling programs collect the data from public sites and convert to text data files
- Global coverage for nav aids and waypoints
- No guarantee that it is always up-to-date



Open data: Airports

- 14480 airports included
- Large airports also feature taxiway and runway layout
- Flight simulator and gaming community have people who collect this information as well



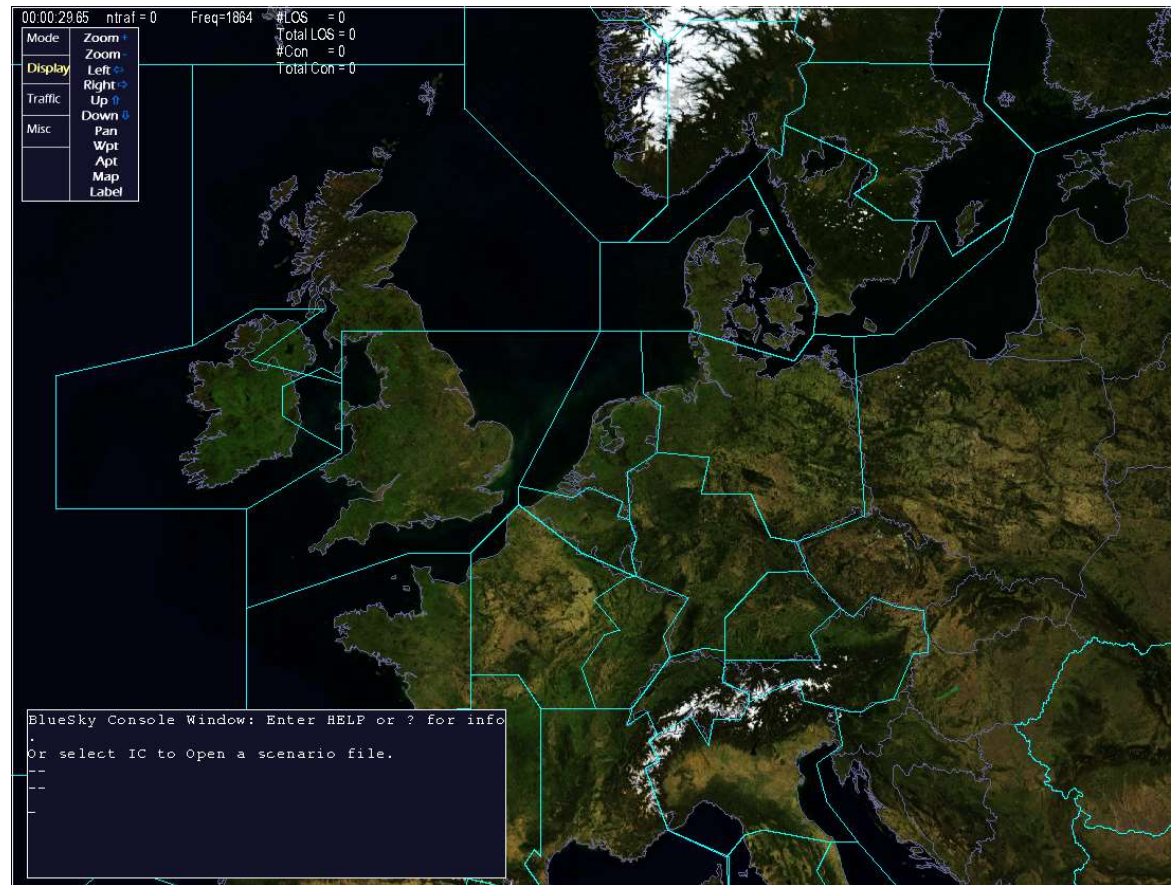
Open Data: Geographical information

- Just background for GUI
- Geo-website contains all required info:
 - Coastlines
 - Rivers
 - Borders
- Satellite image used from Google Earth



Open Data: FIRs, Sectors etc

- Supplied data limited
- Web-crawling possible, no global repository
- Often implemented manually based on scenarios
- Console is not ATCo HMI, but fully reconfigurable
- More help needed



Open Data: Aircraft Performance models

- Compatible Plug& Play with BADA v3.12 (nearly open):
Copy BADA files to empty folder: `.\data\coefficients\BADA`

- AE2100A.xml
- CF6-80C2B1F.xml
- CF6-80E1A2.xml
- CF6-80E1A3.xml
- CF34-8E5.xml
- CF34-10E2A1.xml
- CF34-10E5.xml
- CF34-10E5A1.xml
- CF34-10E6.xml
- CF34-10E6A1.xml
- CFM56-5-A1.xml
- CFM56-5A3.xml
- CFM56-5B53.xml
- CFM56-5B62.xml
- CFM56-5B73.xml
- CFM56-7B24.xml
- CFM56-7B27.xml
- CT7-9B.xml
- JT15D-4 series.xml

- OpenAP: Big Data effort using ADS-B data to develop comprehensive set of aircraft performance files (built-in)
- Generic Open Data models fully based on Open sources (built-in)
- Metrics also use generic measure Energy, independent of Fuel Flow

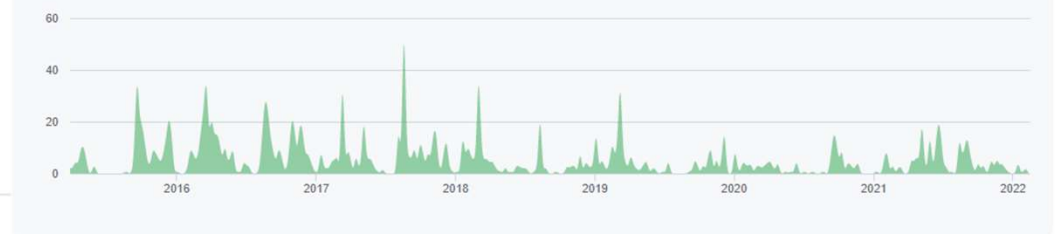
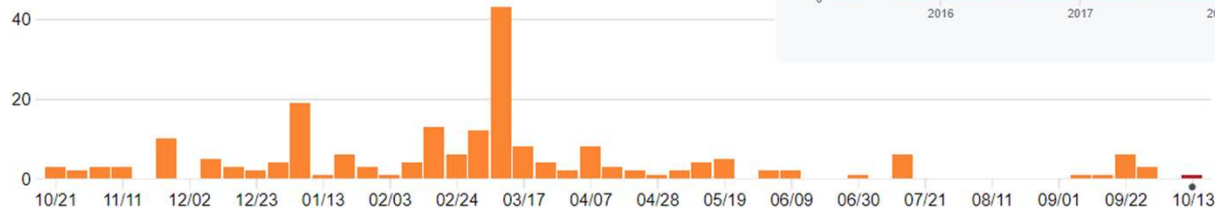
- 0B744.xml
- A319.xml
- A320.xml
- A333.xml
- AT45.xml
- AT75.xml
- B738.xml
- B772.xml
- C550.xml
- D328.xml
- DH8D.xml
- E170.xml
- E190.xml
- F100.xml
- SB20.xml
- SF34.xml



Current status BlueSky community

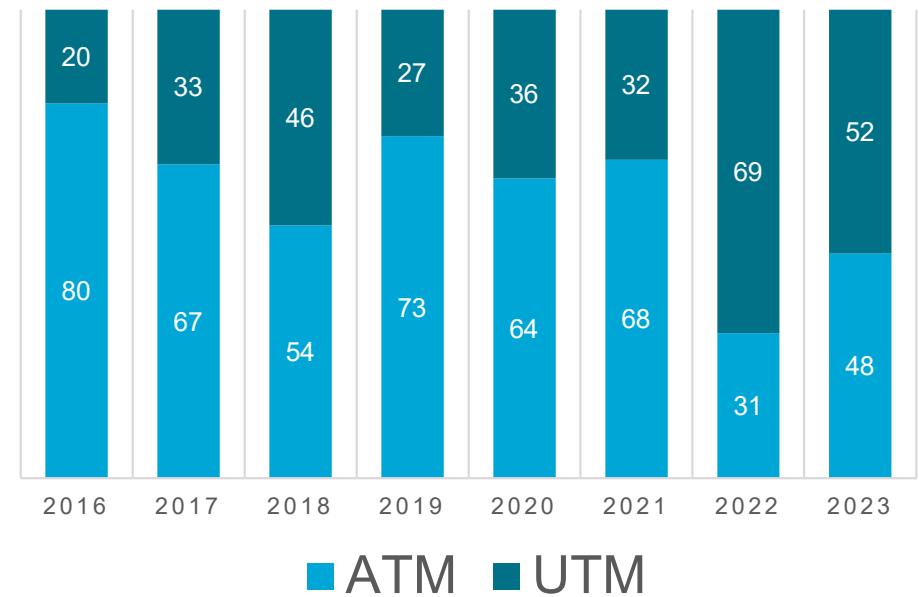
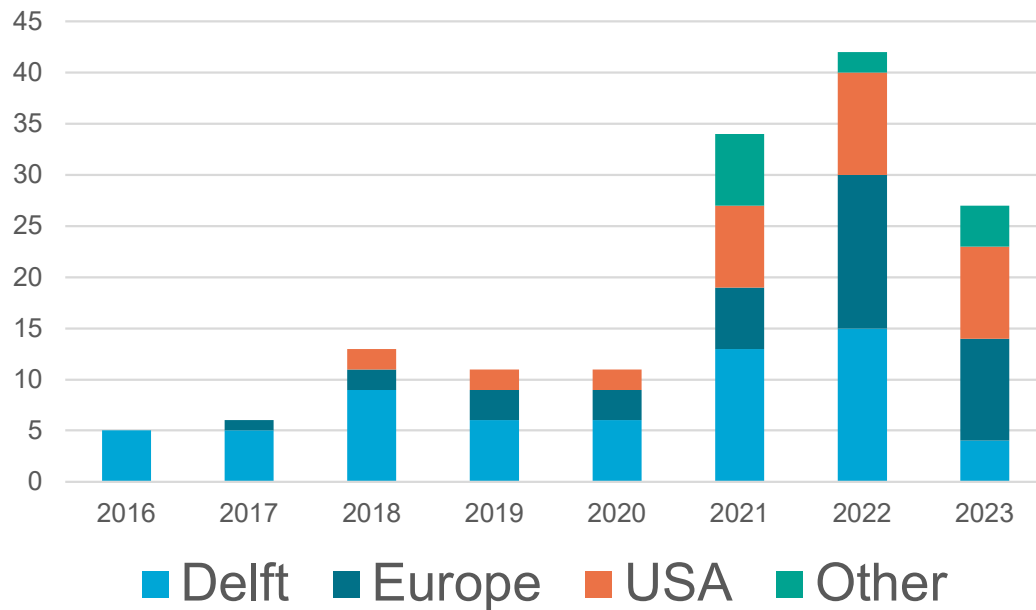
- <https://github.com/TUdelft-CNS-ATM/bluesky/graphs/contributors>
- Global user community (Europe/US/China dominant, but also users in African countries)
<https://github.com/TUdelft-CNS-ATM/bluesky/network/members>
- 100-200 views per day, BlueSky paper cited 100+ x, a.o. because several EU projects use BlueSky
- Many spin-offs, development of plugins (of which 13 have been added to the master version) and 236 forks (and 336 stars, last commit yesterday)

- ± 2400 commits



BlueSky citations over the years

- Status mid 2023



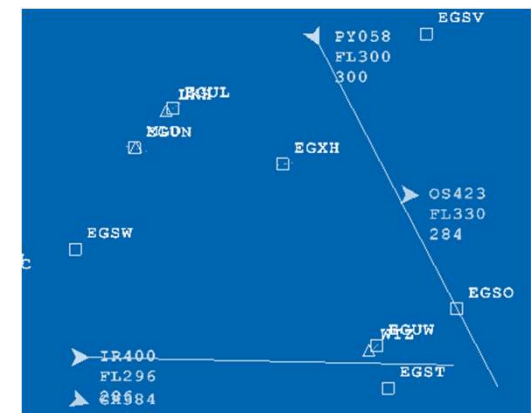
Hoekstra, J. M., & Ellerbroek, J. (2016, June). Bluesky ATC simulator project: an open data and open source approach. In *Proceedings of the 7th international conference on research in air transportation* (Vol. 131, p. 132). USA/Europe: FAA/Eurocontrol.

Spin-off: OpenAP models

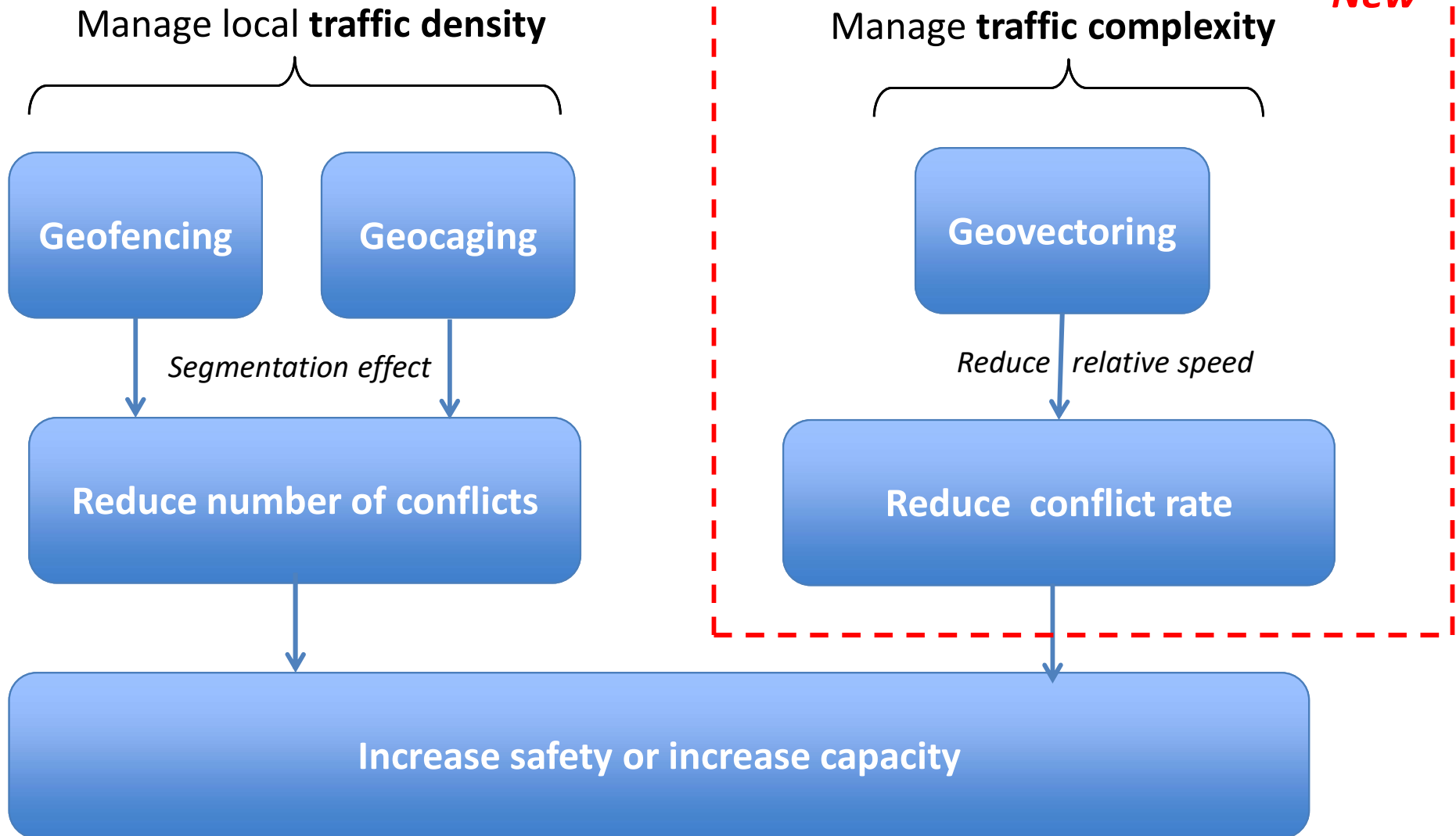
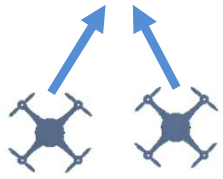
- Based on Bayesian approach (particle model) to ADS-B data
- Also supplied with BlueSky (Options: BADA/OpenAP/Legacy)
- Structure is not too different from BADA but no license required. Physics based, focus is on performance limits.
- Includes generic drone models and equivalence table
- Can also be used in other modules or programs
- <https://github.com/TUdelft-CNS-ATM/openap>

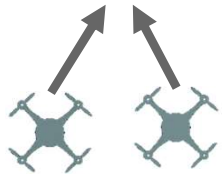
Applications so far

- **Airspace and Complexity metrics** in existing traffic demand scenarios
- Simulate/analyze ATM procedures like **Upstream delay absorption**
- **AMAN and XMAN** interference of scheduling with pop-up traffic
- **Visualize ADS-B** data from receivers or networks
- **Conflict Detection and Resolution/ Detect & Avoid** studies
- **ASAS/Swarming** algorithms
- Effect of procedures and **airspace structure** on capacity and safety (**UTM**)
- **Machine learning** plug-ins for sequencing task



Manage high density U-space



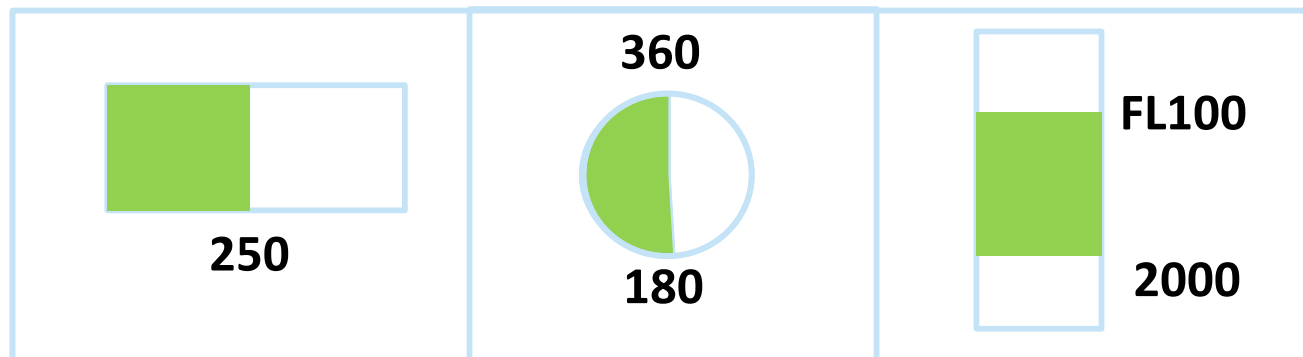


Geovectoring: a generic way to define this

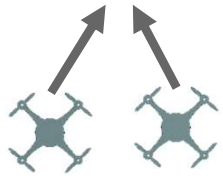
- Define an area (polygon/min alt/max alt) and define a **geovector** consisting of intervals in three specific dimensions:

$$V_{geo} = \begin{pmatrix} [Groundspeed_{min}, Groundspeed_{max}] \\ [Course_{min}, Course_{max}] \\ [VerticalSpeed_{min}, VerticalSpeed_{max}] \end{pmatrix} = f(lat, lon, altitude)$$

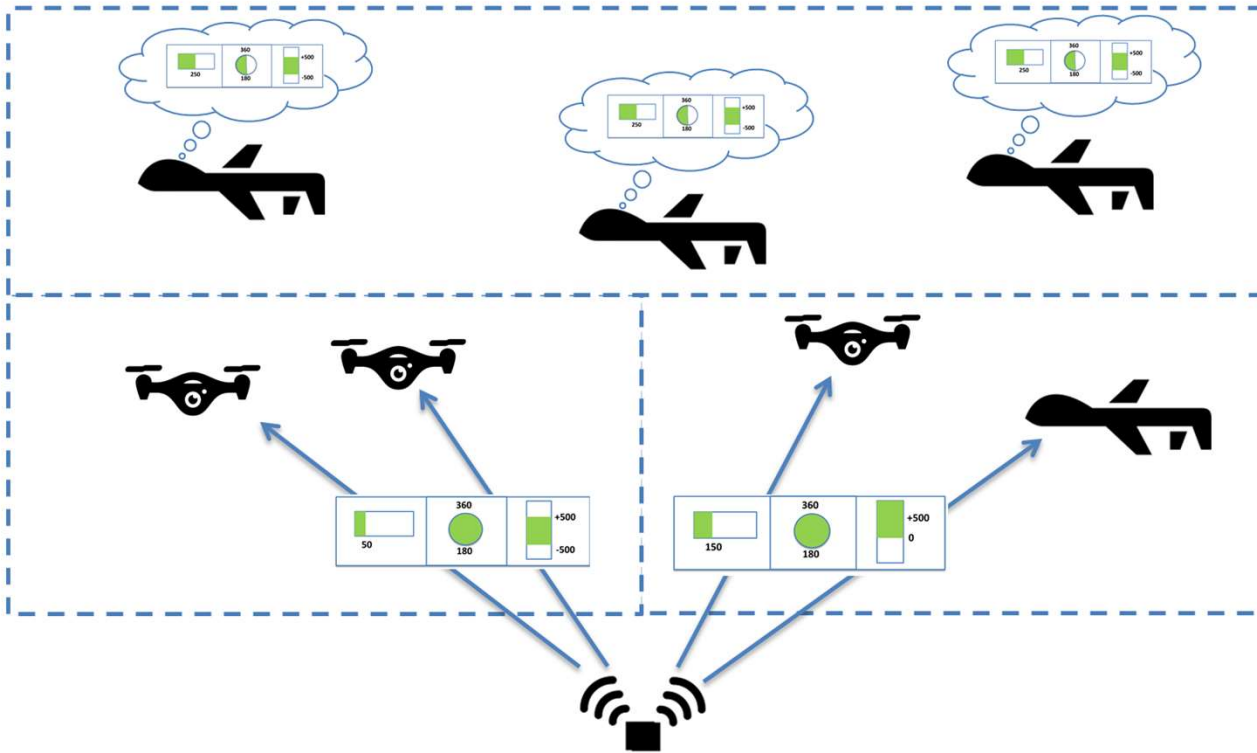
Spd
Hdg
V/S



- Note: horizontal speed is defined in polar notation: (GS , HDG)



Static vs Dynamic Geovectoring



Static
of navigation/procedural
data for a given airspace
can be stored or uplinked

Dynamic
s with time and situation
Needs to be uplinked

It's like Wikipedia: your help is needed

- Fully open source, so help us:
- **Missing data in GUI/model: sectors, airspaces, weather/wind data**
- **Debugging:** send reports
- Extension: **request us to add functionality** you need
- Extension: add your own **functionality in a plug in**
- Wikipedia philosophy: open source and forever “beta”
- Documentation:
<https://github.com/TU Delft-CNS-ATM/bluesky/wiki>

The End Goal of BlueSky



- A fully open source, open data, with extensive features but still easy to use and develop by everyone without any restrictions or licensing
- Exchange metrics (common measures)
- Exchange scenarios
- Stand on the shoulders of your fellow researchers (giants?)
- Join the BlueSky community

