The FlightSchedule Profiler

An attempt to synthetise visually an airport's flight offer in time and space.

Jean-Yves Blaise, Iwona Dudek

The challenge: providing users with a clear-cut visual overview of an airport's flight offer.
Our objective: Not *replacing* but *complementing* reservation systems in upstream tasks.

## 1 The Issue - limitations of flight reservation systems

There definitely is a lot of information available on the net through airlines reservation systems, airports destination maps, *etc.* But is that '*space + time*' information really synthetic, easy to read ? The very nature of the query one fires on such systems (typically "*find flights from A to B on day D, order them by price / duration*") strongly impacts the type of information users will retrieve, and the way it will be displayed.

Shortly said, such queries end up on:

- **sequential data series** - *one flight after the other, page per page*
- **verbose presentations** - *time slots and routes are given as texts, temporal densities not assessed*
- **focus views** - *details on one flight at a time*
- **space parameter limited to a textual list of airports*

- Self-proclaimed information retrieval - basically a mess,
- Too much (irrelevant) data,
- No overview, no user-side mental image of time + space alternatives,
- Lacks ... infoVis?

2 Are there alternatives ? *Inspiring designs from the golden age**

* A tribute to M. Friendly's "The Golden Age of Statistical Graphics"
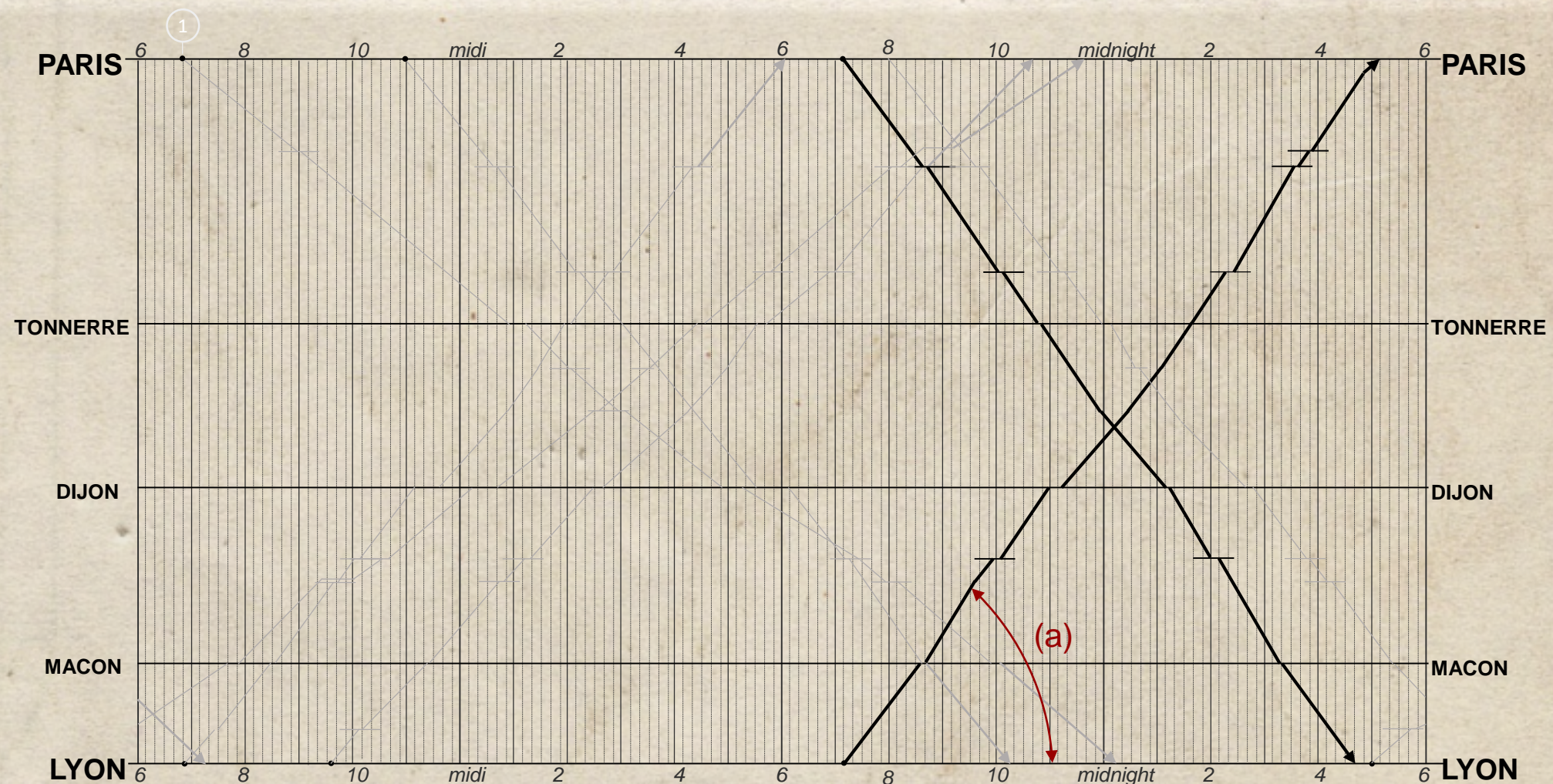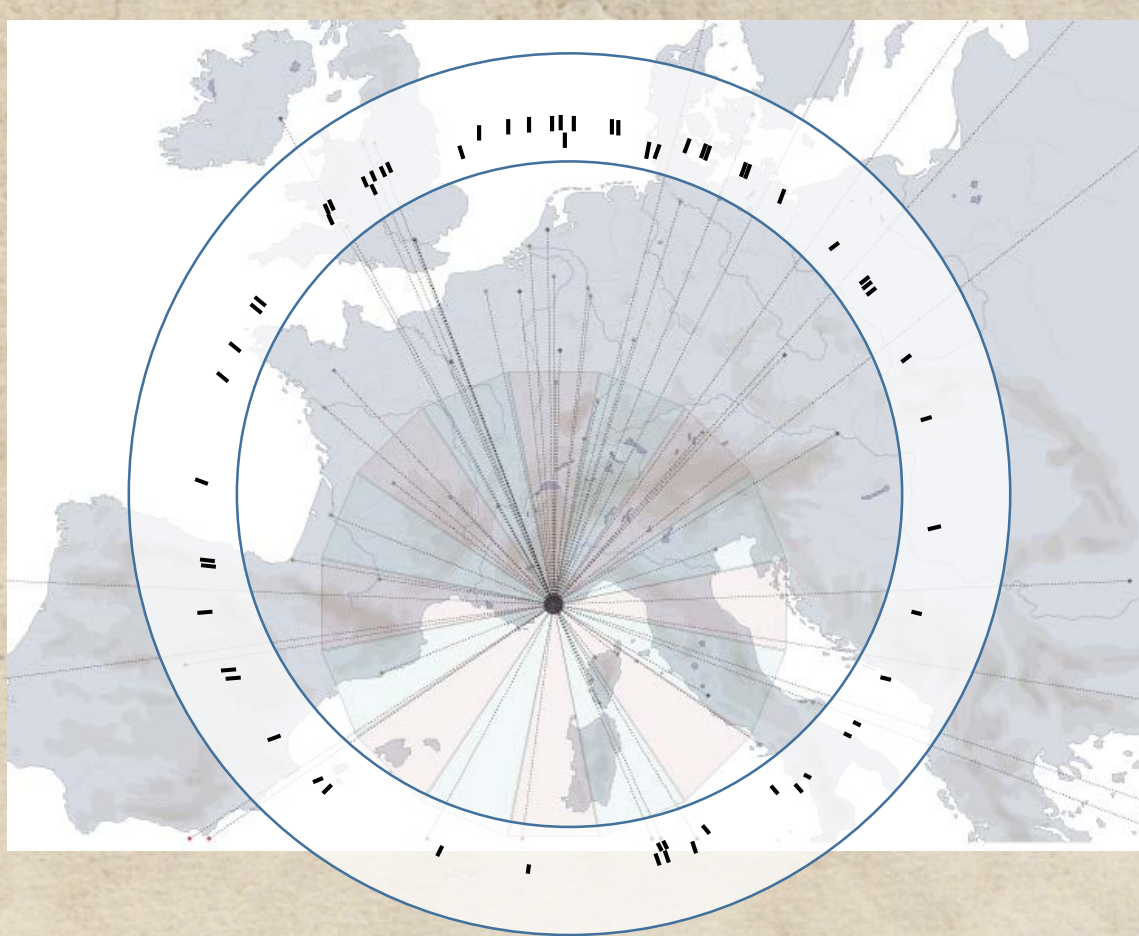

Etienne Jules Marey's 1885 train schedule (redrawn and simplified). Time runs horizontally from left to right (24 hour slot, starting 6AM). Railway stations (cities) are distributed vertically.

Oblique lines corresponds to trains connecting Paris to Lyon and vice-versa. The angle (a) represents the speed of trains (the duration of the travel in fact). Oblique lines redrawn in black correspond to the fastest trains – the structure of the graphics makes it so that it is simply straightforward to uncover a pattern: these trains depart at the same time.
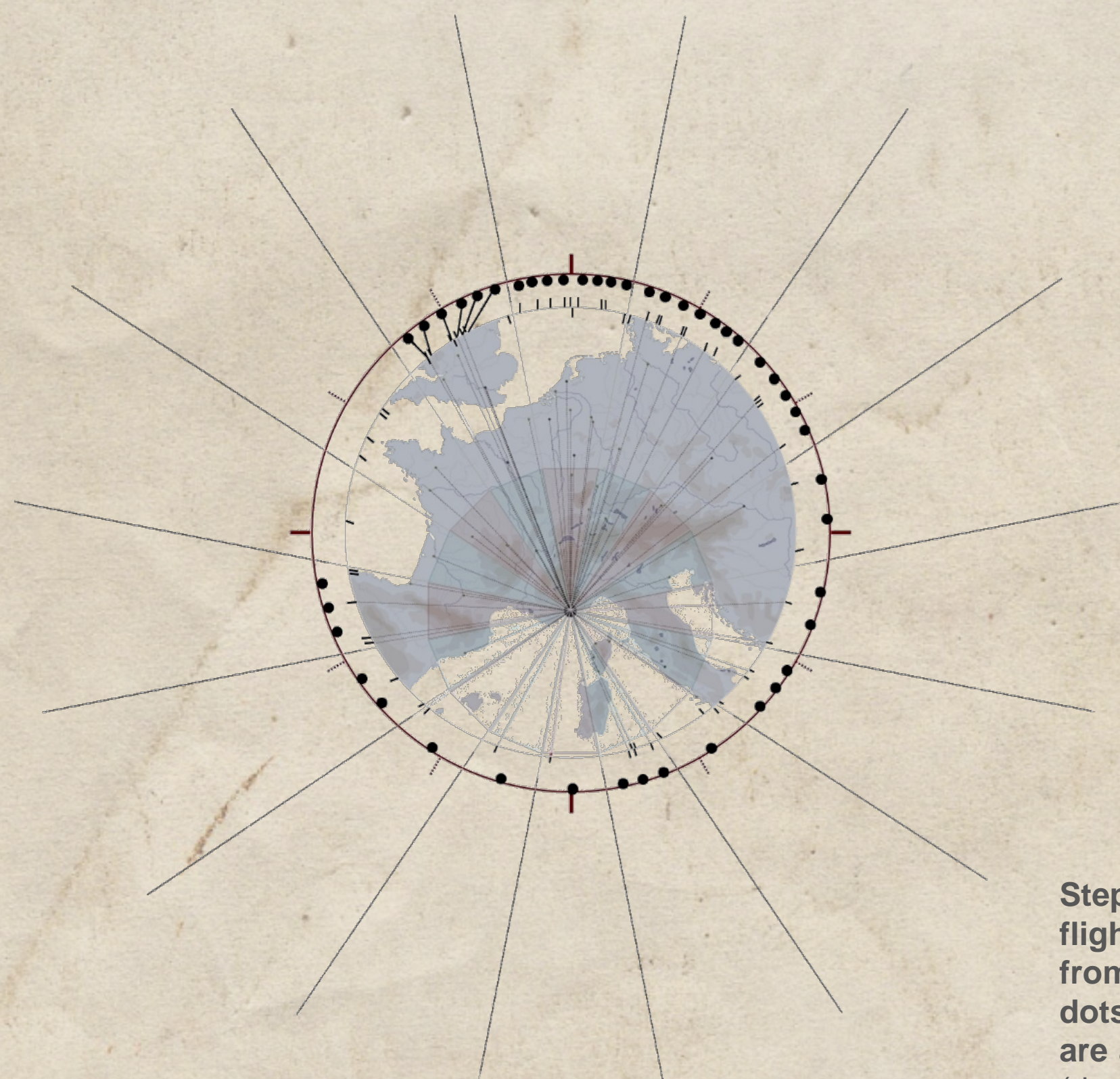
Note that this graphics helps spotting at a glance for instance which of all trains is the slowest (circle "1" above the top horizontal line), or lets you compare the stopover time for each train, in both direction, in all railway stations.

3 The 'FlightSchedule' proof of concept visualisation - investigating the way travel data can be displayed visually.

**Three key choices:** *stylized cartography, ordinal time model, "details on demand"*

- ● regular flight (every Monday over the period)
- ● seasonal or occasional flight
- ● destination available on Mondays
- ● destination available on other days of the week

Step 1 : The cartographic background. Origin-destination vectors intersecting a circle

Step 2 : Redistributing destinations along a second, exterior circle so as to avoid overlapping.
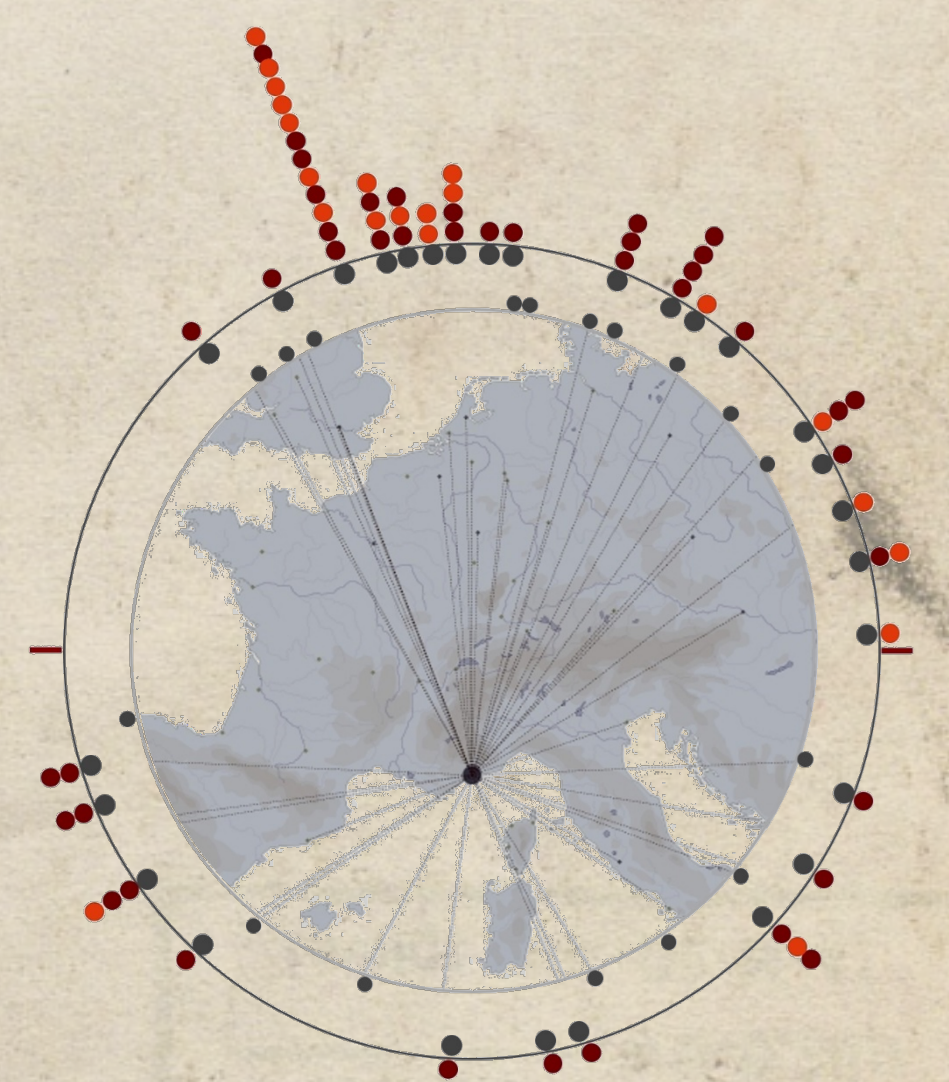
Step 3 : Introducing the ordinal time layer (Monday flights) - each coloured dot corresponds to one flight from Nice to a destination airport. "Piles" of coloured dots correspond to destinations for which several flights are available on Mondays, sorted by chronological order (details available on user demand).

Note the varying proportion of non-regular flights between those heading north-west and others. User-side interaction triggers the displaying of textual and visual information about cities, airport codes, place types, departure and arrival times, operating periods etc.
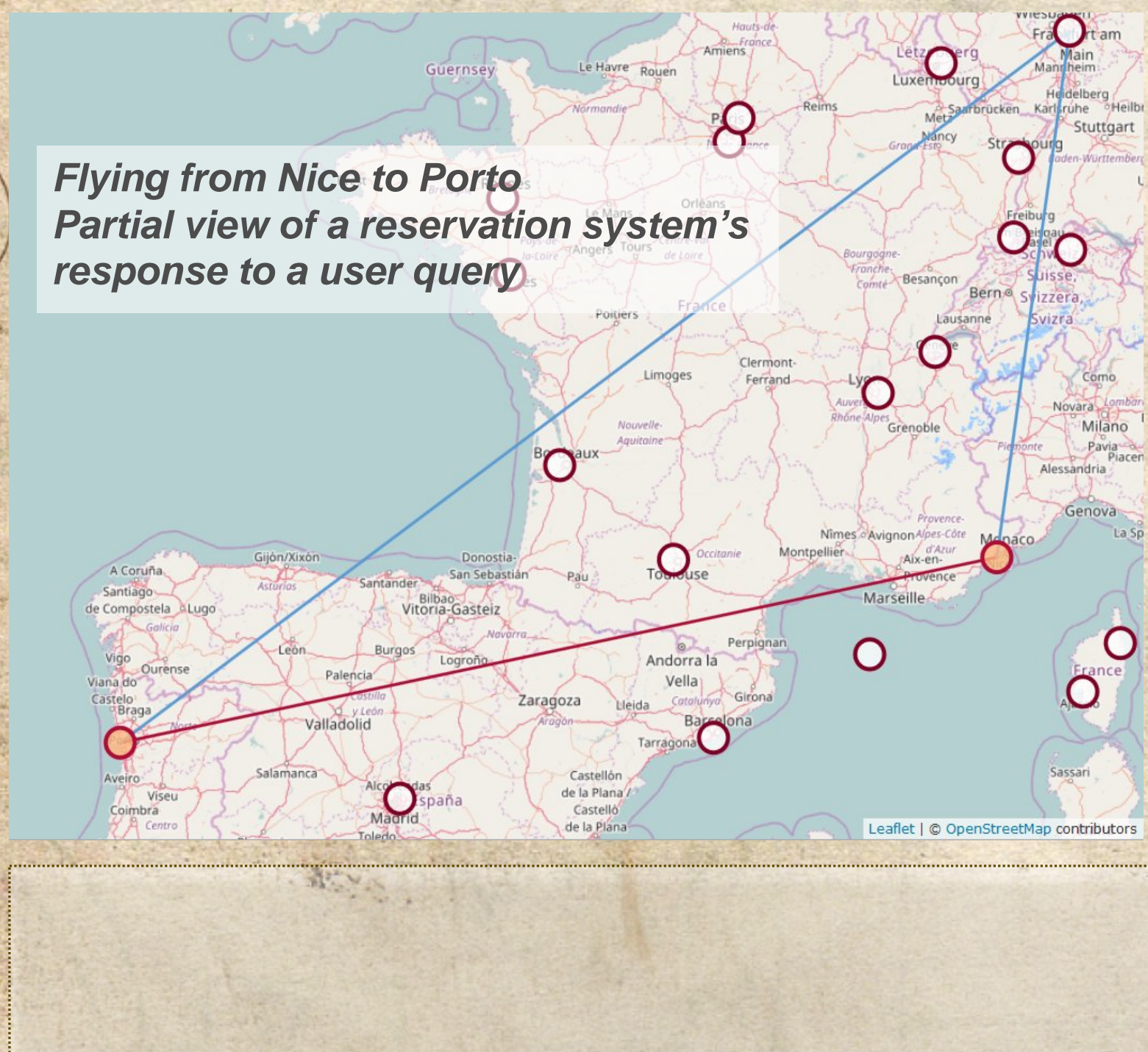
The FlightSchedule Profiler

An attempt to synthetise visually an airport's flight offer in time and space.

Jean-Yves Blaise, Iwona Dudek

The challenge: providing users with a clear-cut visual overview of an airport's flight offer.
Our objective: Not *replacing* but *complementing* reservation systems in upstream tasks.

1 The Issue - limitations of flight reservation systems

Flying from Nice to Porto
Partial view of a reservation system's response to a user query

Leaflet | © OpenStreetMap contributors

2 Are there alternatives ?
Inspiring designs from the golden age*

* A tribute to M. Friendly's "The Golden Age of Statistical Graphics"
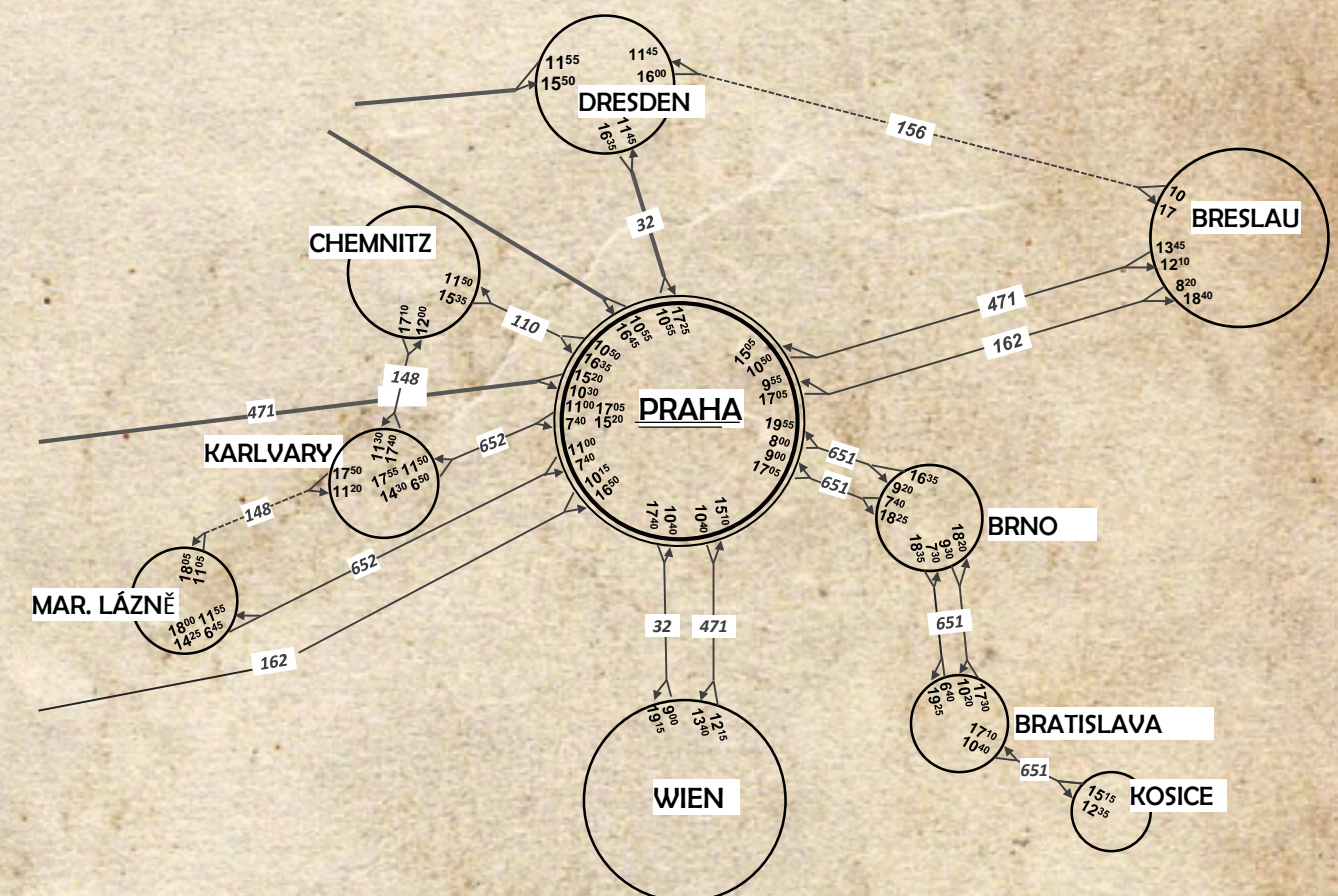
An extract of the 1933 Czech Airlines map and schedule (redrawn from E.R Tufte, Envisioning information, relative sizes of circles preserved from the original). **Each circle corresponds to relative locations in space; departure and arrival times are shown inside the circles, lines with flight codes connect airports (the presence of an arrow differentiates inbound from outbound flights). Brown circles (top left) correspond to the "real" location of cities on a geographic map** .

This design combines in one unique visualisation parameters *space* (position of cities) and *time* - departure / arrival times.

There are here readability concerns that could be pointed at – the orientation of departure / arrival times inside circles for instance – but it shows there definitely is room for graphic creativity in the context of travel data.

3 The 'FlightSchedule' proof of concept visualisation - investigating the way travel data can be displayed visually.

Extending the experimentation with discrete time

5 PM
4 PM
3 PM
t
6-12 h
12-21 h
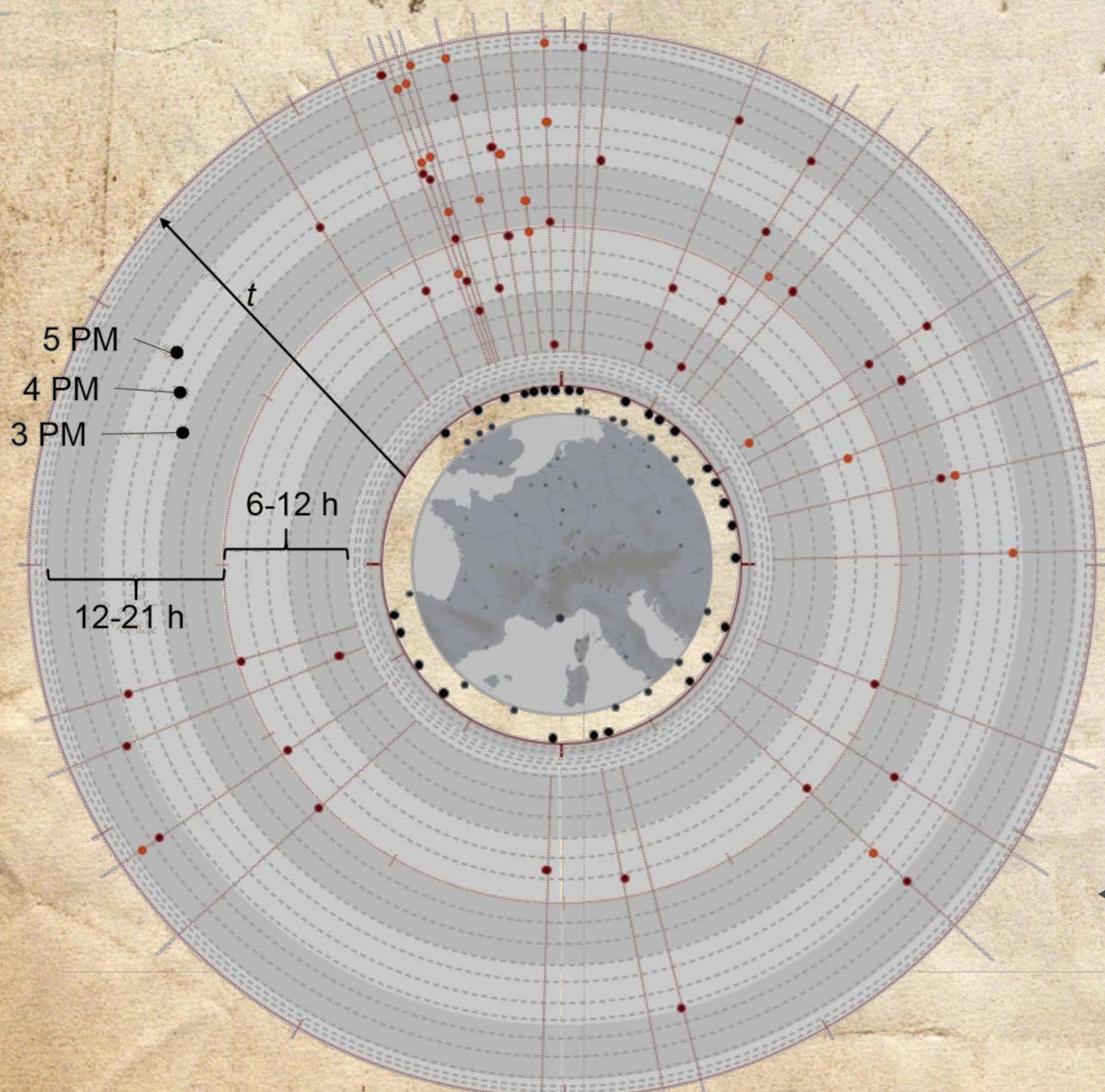
Focus view on a Tuesday, one flight highlighted: the green arc show the duration, the red arrow positions the flights in the day (24 hours, discrete time) while the white arrow positions the flight in a clock-like representation of a 12-hours slot.

Munich (MUC)
13:25 - 14:55
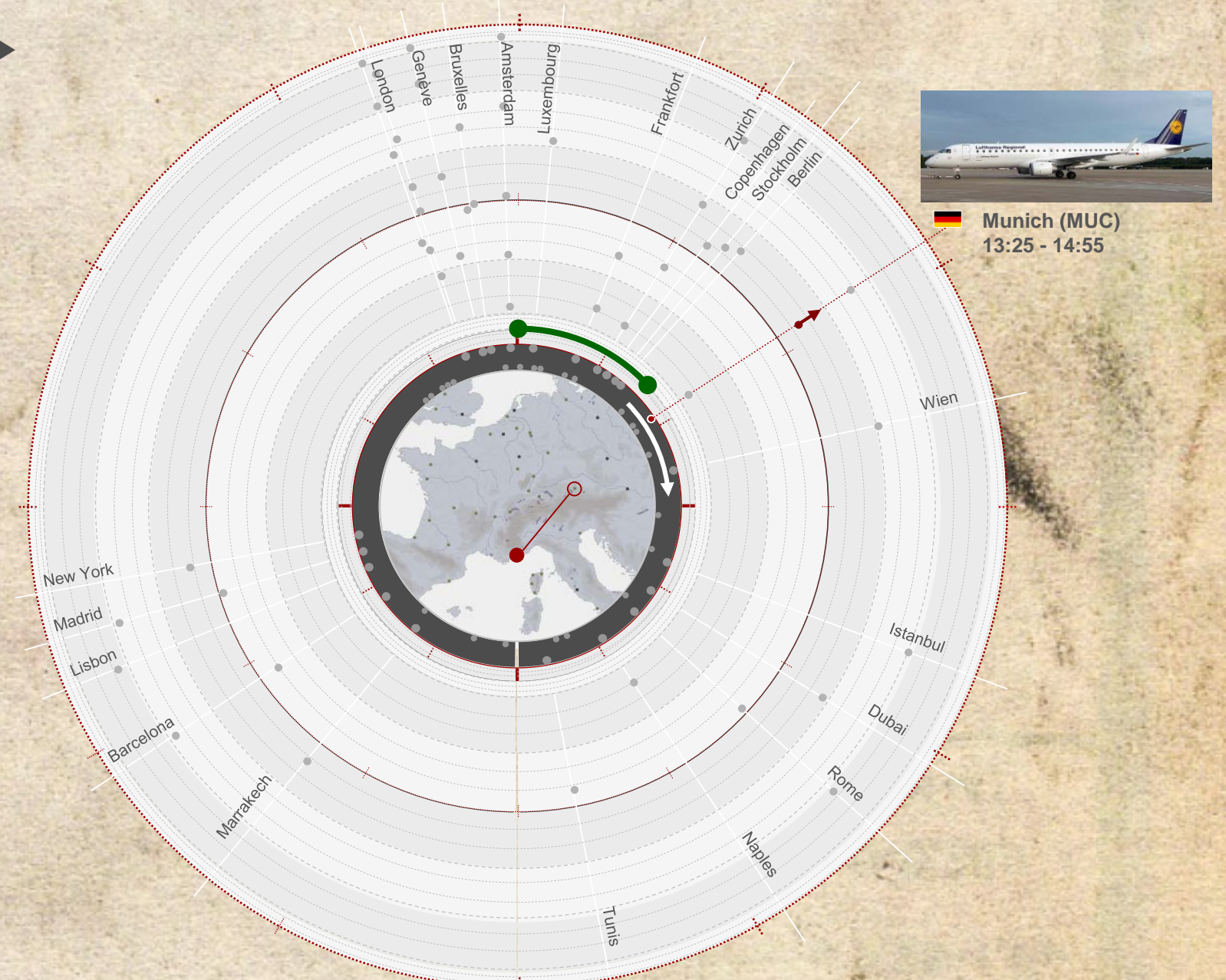
◄ The discrete time layer (Monday flights): concentric circles correspond to hours of a day (time runs outwards). Each dot corresponds to a given flight, with the same colour codes as in the ordinal time layer.

The visualisation is an attempt at assessing temporal densities throughout the day.

The FlightSchedule Profiler

An attempt to synthetise visually an airport's flight offer in time and space.

Jean-Yves Blaise, Iwona Dudek

The challenge: providing users with a clear-cut visual overview of an airport's flight offer.
Our objective: Not *replacing* but *complementing* reservation systems in upstream tasks.

1 The Issue - limitations of flight reservation systems

Flying from Nice to Porto
Partial view of a reservation system's response to a user query
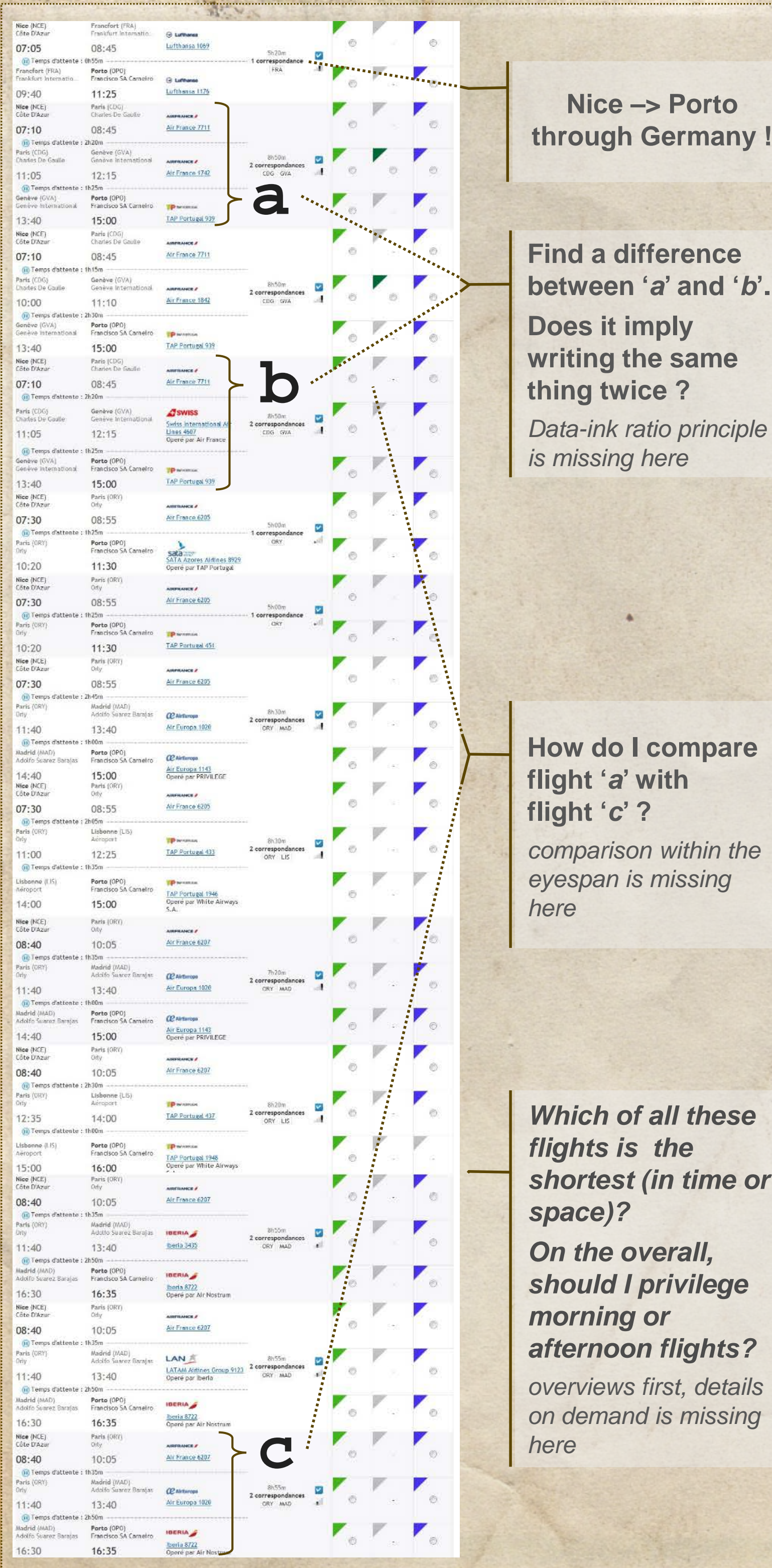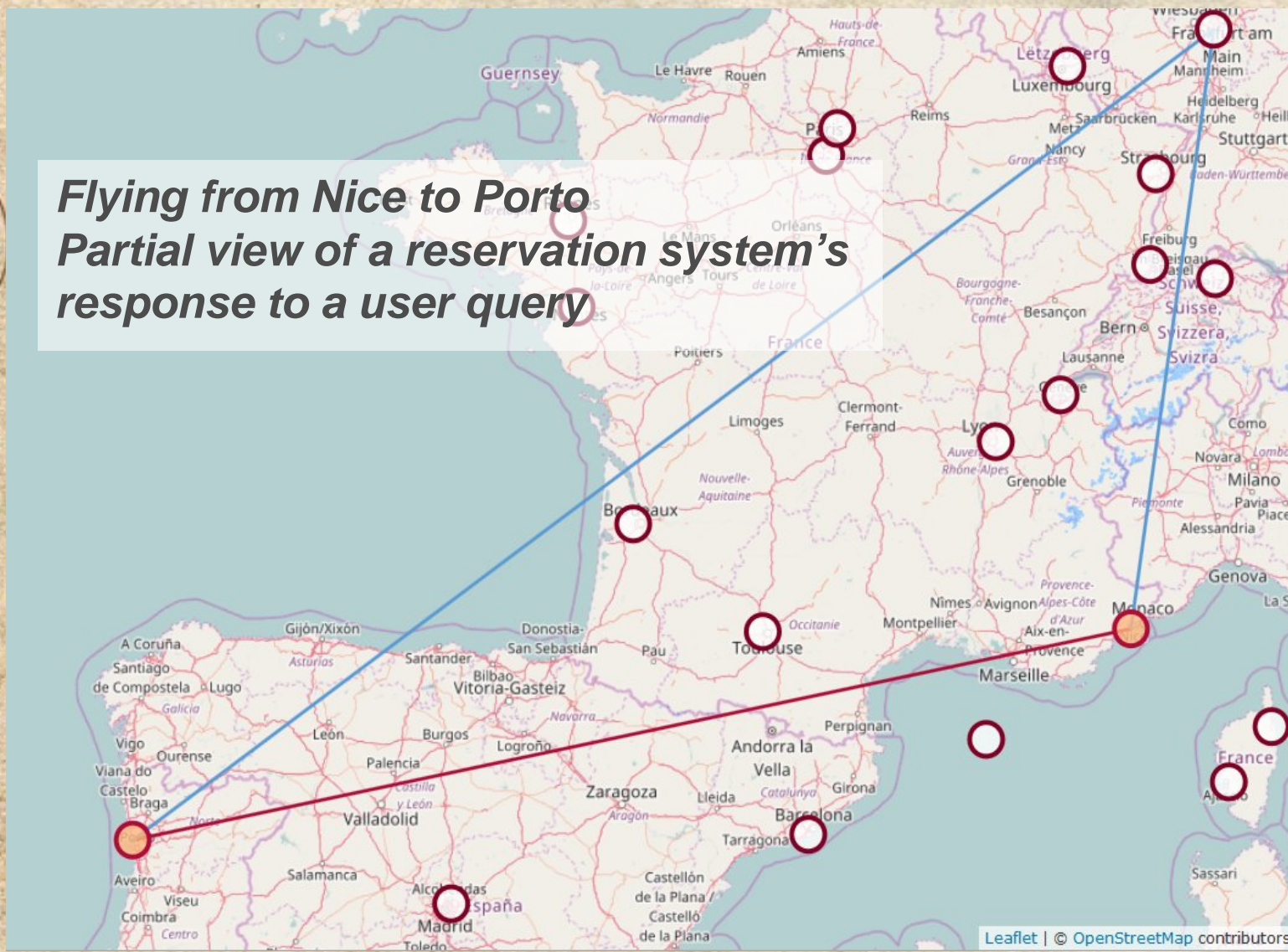
Nice –> Porto through Germany !

a

b

Find a difference between '*a*' and '*b*'.

Does it imply writing the same thing twice ?

Data-ink ratio principle is missing here

How do I compare flight '*a*' with flight '*c*' ?

comparison within the eyespan is missing here

Which of all these flights is the shortest (in time or space)?

On the overall, should I privilege morning or afternoon flights?

overviews first, details on demand is missing here

c