Welcome to El Niño!

El Niño – Welcome

EL NIÑO

Welcome to our office!

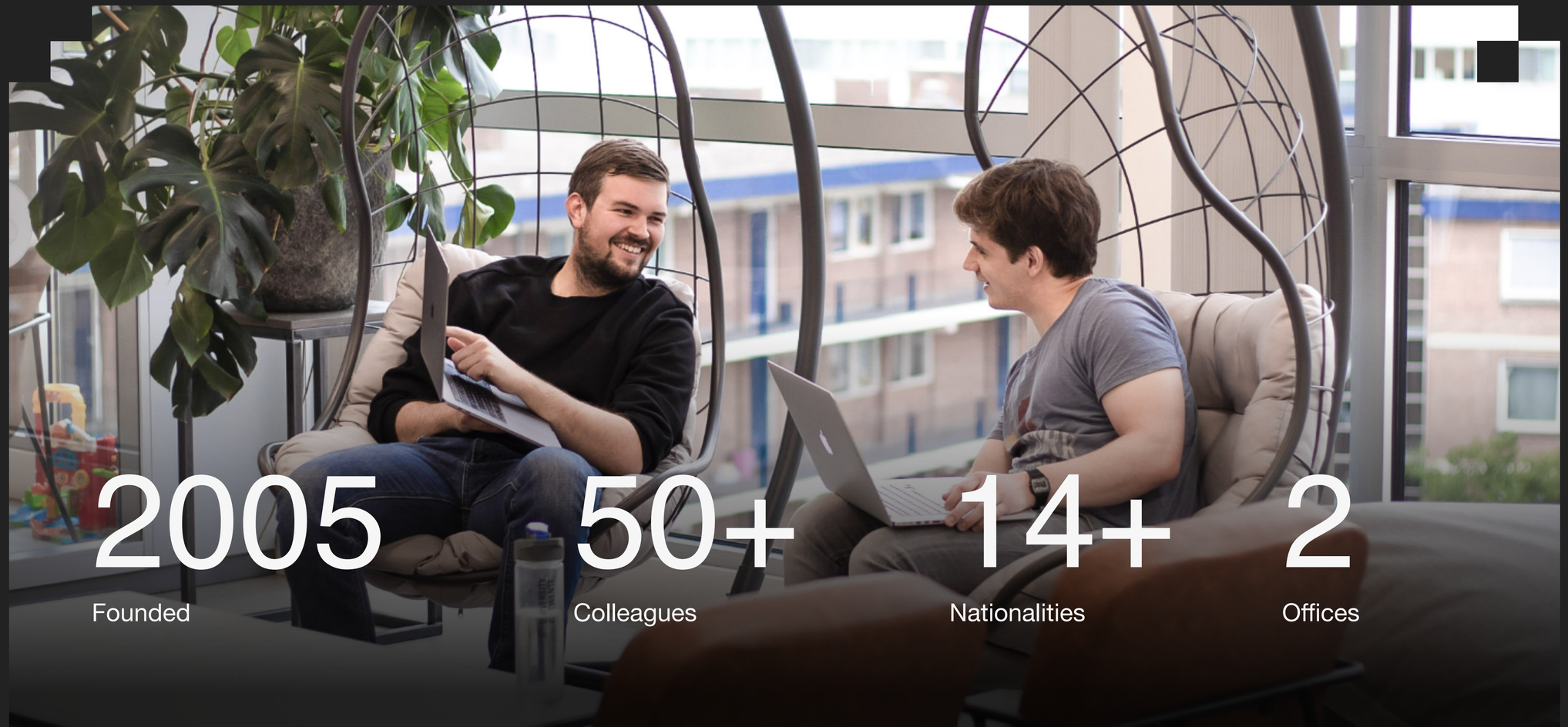
I'll introduce you to El Niño

After that, we enjoy the Web Development Meetup event!

I'll stay at the event, if you have any questions, don't hesitate to talk to me!

Our numbers

El Niño At a glance



We are crazy about technology and with our knowledge and passion we help others succeed

What drives us

Kicking ass in digital development

Whether it is a mobile app, web application or webshop: we love a good challenge.

If it was easy, everybody would do it. All our projects are technology driven and we always try to find the best solution possible. But whatever the solution we come up with, we strive for quality and go for the best experience for the end user.



Services we provide

Custom web applications

Automating business processes through developing custom web applications. Our software solutions are scalable, flexible
and secure.

Webshop development

Choose for online success with the eCommerce platform that best suits your business. From Magento and Saleor to Shopify, always developed especially for your customers.

App development

Delivering the most powerful and user-friendly mobile experiences with native iOS and Android apps.

Data science

Don’t just take our word for it. See what the data says. Let’s start using data to achieve your goals through strategic advice and practical implementation.

Design

User-centered design for digital solutions that help businesses enhance the value they provide.

Some of our clients

victron energy
BLUE POWER

A Dutch company specialized in the design and manufacture of products for battery-based energy systems. Since 1975, grown to be an inter-national enterprise with nearly 1,000 products sold worldwide.

JUST EAT Takeaway.com

A leading global online food delivery marketplace, connecting consumers and restaurants through its plat-forms. With over 692,000 connected partners offer-ing consumers a wide variety of food choice.

FrieslandCampina

Friesland Campina provides millions of people all over the world with dairy products. With 21,715 employees and branches in 31 countries, their products find their way to more than a 100 countries.

Segway-Ninebot

Global enterprise in the field of intelligent short-distance transportation and service robots. The company’s businesses are inter-national, selling products in more than 80 countries and regions.

nedap

Nedap aims to develop technology that helps people to be more productive and successful in their work with 750+ employees spread over 10 offices worldwide.

TECNOTION

Independent supplier of linear- and torque motors. They offer motor technology to place in motion solutions. Tecnotion has a global presence, with production plants in The Netherlands and China.

ZorgSelect

This largest regional job board, helps Eastern Netherlands' healthcare institutions find staff. Coupled with E-Portfolio, a digital resumé for healthcare workers, it offers a comprehensive healthcare career development solution.

kijkshop

Kijkshop used to be a store offering an assortment of non-food products nationwide. Now they aim to be the No. 1 live shopping platform in the Netherlands, soon to be launched.

Feel free to contact us

if you need any further information

info@elnino.tech

053 820 0285

EL NIÑO

Enjoy the
food and drinks

Up next: Han van der Veen
Building a public transport route planner for the Swedish Railways

Feedback?
Share it with us!

sqills

Building a Route Planner

Building a public transportation route planner for Swedish Rail

Han van der Veen



Introduction



Han van der Veen

Software Engineer @ Sqills



S3 Passenger

Search travel options

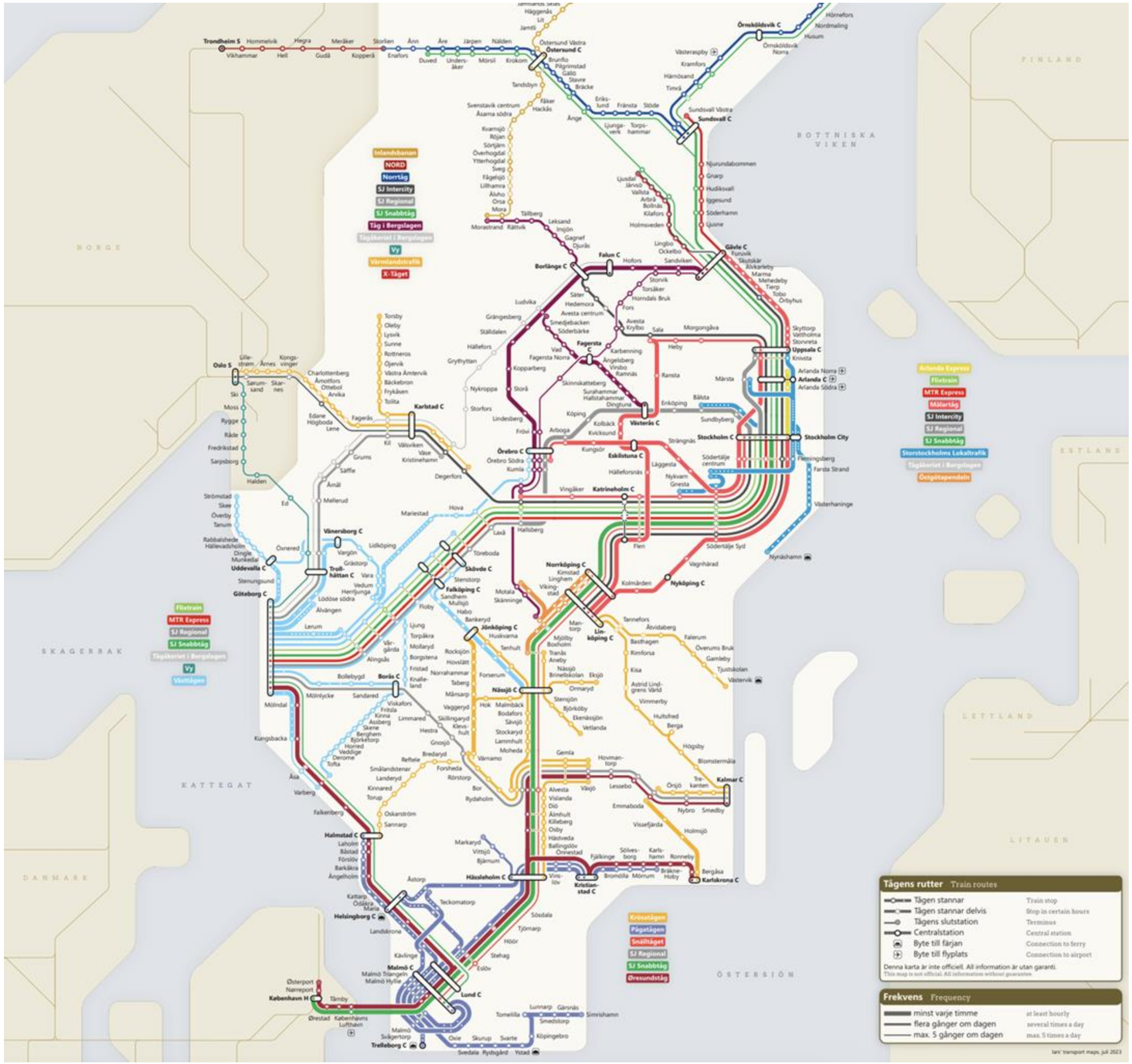
Find prices

Book Tickets



S3 Navigator


Snälltåget
NOR WAY Busekspress
SJ
Rail Delivery Group
National Rail
DSB
Iarnród Éireann
Irish Rail
B
benerail
eurostar
VIA Rail Canada
Otpan Logistic
renfe
SNCF
ter
itabus
iryo
BlaBlaBus
.italo
OUIGO
JUNIOR
& Cie
brightline



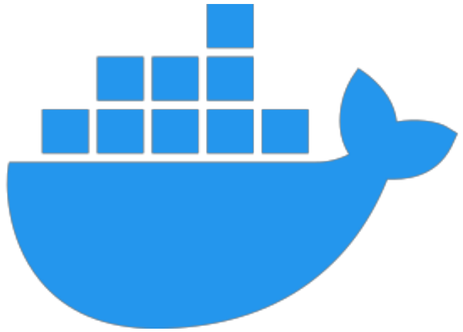
88k departures a day
50m connections in memory

Global Architecture

Technical Challenges

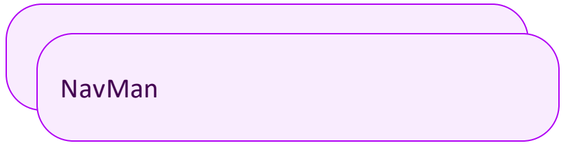






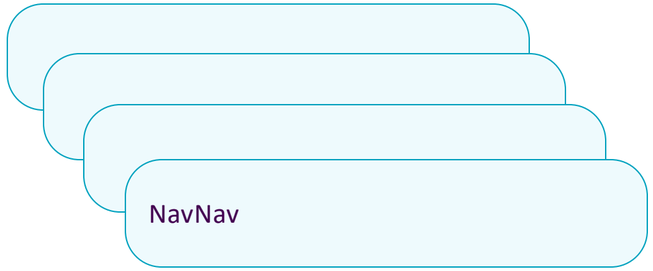
On JVM

Global Architecture



Management

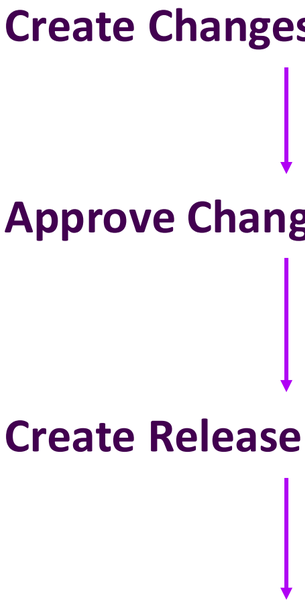
Data management
Batch jobs
Background processes
UI



Navigation engine

Lookup APIs
Navigation APIs

Workflow for Management




```
git branch [changeset]

git checkout main
git merge [changeset]

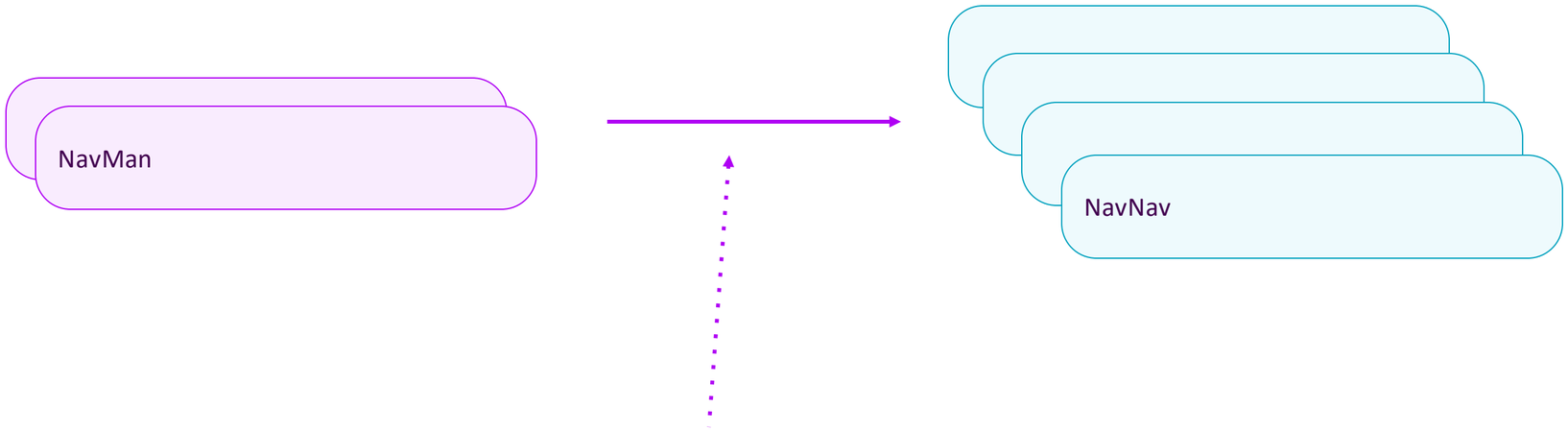
git tag [merge-commit]

git push [tag]
```

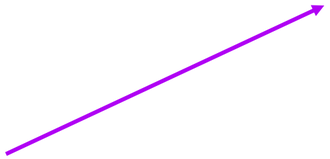
Global Architecture

Technical Challenges

Global Architecture

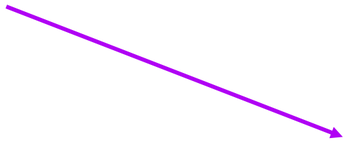


ID	TrxID	Data
UTR	1	Utrecht
ENS	2	Enschede
ENS	3	Enschede Updated
ZWO	4	Zwolle
AMS	5	Amsterdam
UTR	6	Utrecht Updated



Release at 3

ID	TrxID	Data
UTR	1	Utrecht
ENS	3	Enschede Updated



Release at 6

ID	TrxID	Data
UTR	6	Utrecht Updated
ENS	3	Enschede Updated
ZWO	4	Zwolle
AMS	5	Amsterdam

Source is at release 3

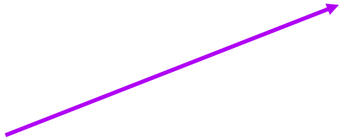
ID	TrxID	Data
UTR	1	Utrecht
ENS	3	Enschede Updated

Update to 6



Target release 6

ID	TrxID	Data
UTR	6	Utrecht Updated
ENS	3	Enschede Updated
ZWO	4	Zwolle
AMS	5	Amsterdam



New

Improving Performance by reducing allocations

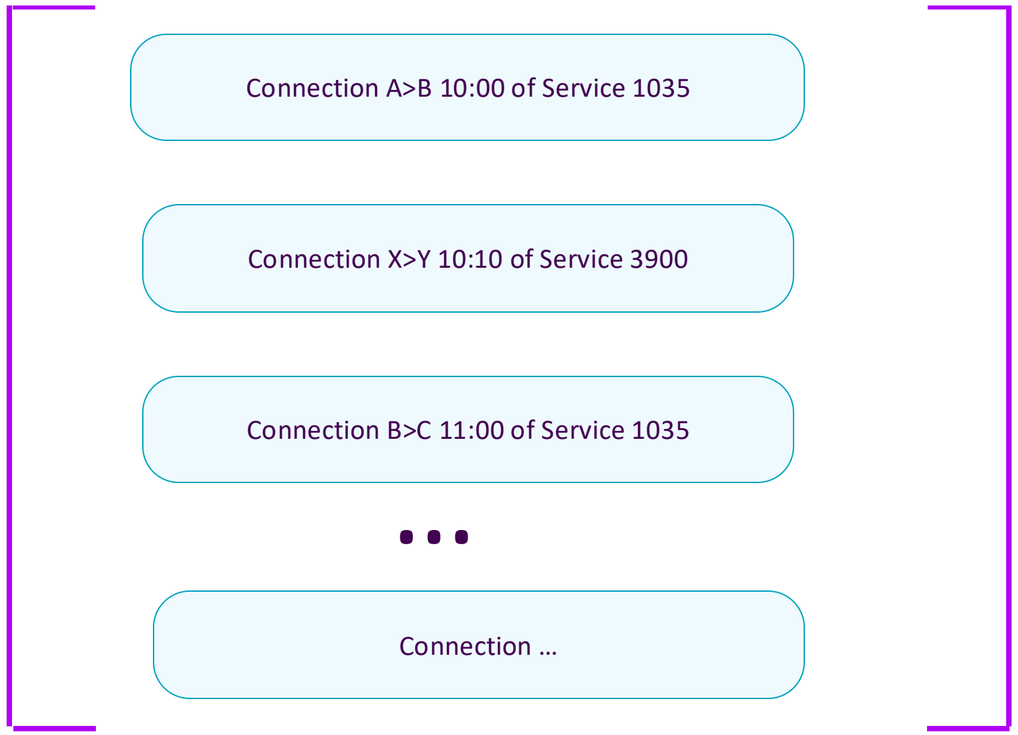
Improving Memory Usage by reducing object sizes

Connection Scan Algorithm

Phase One: Explore what is reachable (physically)

Phase Two: Apply business rules and create profiles

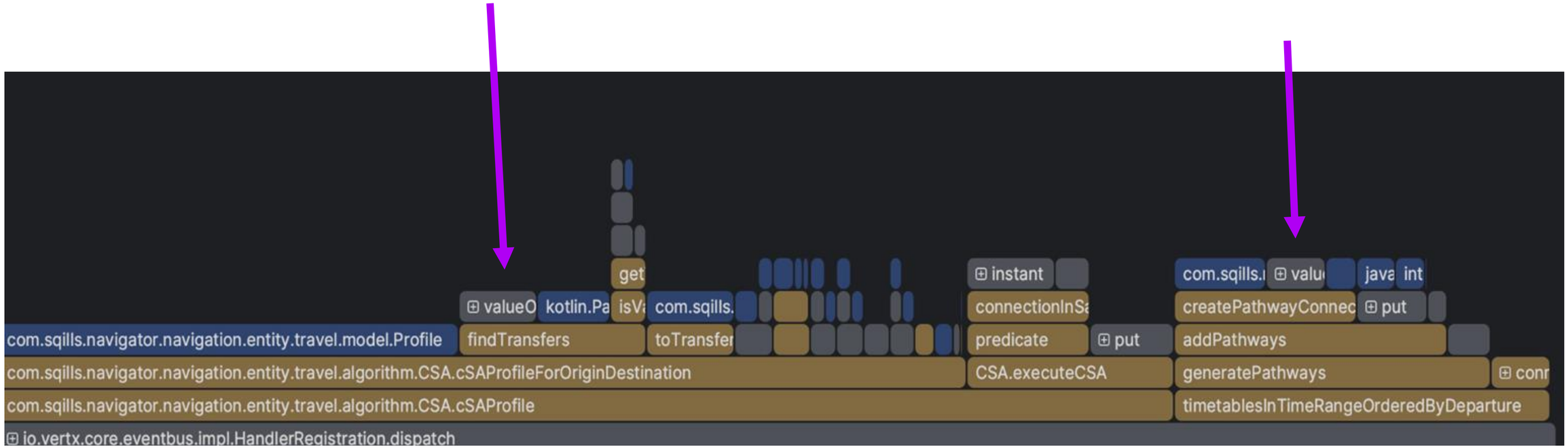
A profile contains everything to reach the destination given a travel path on a given station and time.



Sorted by departure time

https://arxiv.org/pdf/1703.05997

Object Allocation Flame Graph



Auto boxing

```
val value: Int = 10 // using the java int primitive under the hood

val valueNullable: Int? = 10 // using Integer object under the hood
```

Worse when using generics

```
val map = HashMap<Int, Long>()
map[1] = 10L

// under the hood becomes
map[Integer.valueOf(1)] = Long.valueOf(10)

// but also every time you get something from a hash map
val result = map[Integer.valueOf(1)] // map[1]
```

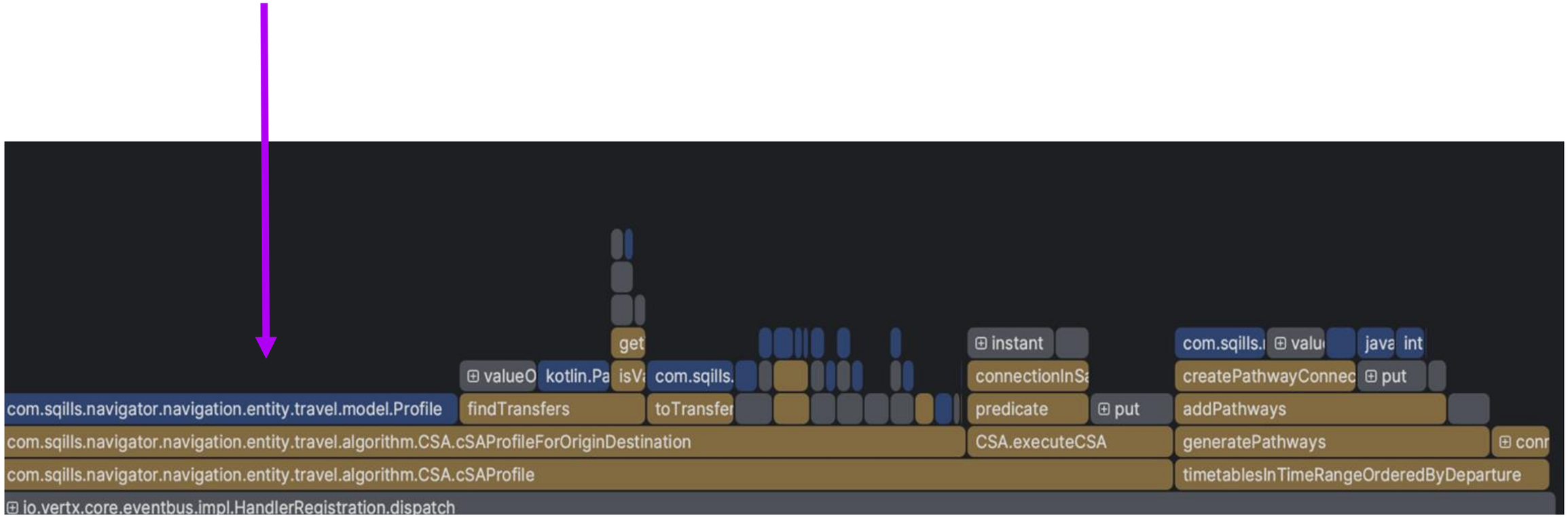
Every .valueOf is a new objection allocation

Auto boxing

```
val map = TIntLongHashMap()
map[1] = 10L // is only using primivites
val result = map[1] // only using primitives.
```

Performance gain of over 50%!

A lot less work for the GC


get
instant
com.sqills.
valu
java
int
valueO
kotlin.Pa
isV
com.sqills.
connectionInSa
createPathwayConnec
put
com.sqills.navigator.navigation.entity.travel.model.Profile
findTransfers
toTransfer
predicate
put
addPathways
com.sqills.navigator.navigation.entity.travel.algorithm.CSA.cSAProfileForOriginDestination
CSA.executeCSA
generatePathways
conr
com.sqills.navigator.navigation.entity.travel.algorithm.CSA.cSAProfile
timetablesInTimeRangeOrderedByDeparture
io.vertx.core.eventbus.impl.HandlerRegistration.dispatch

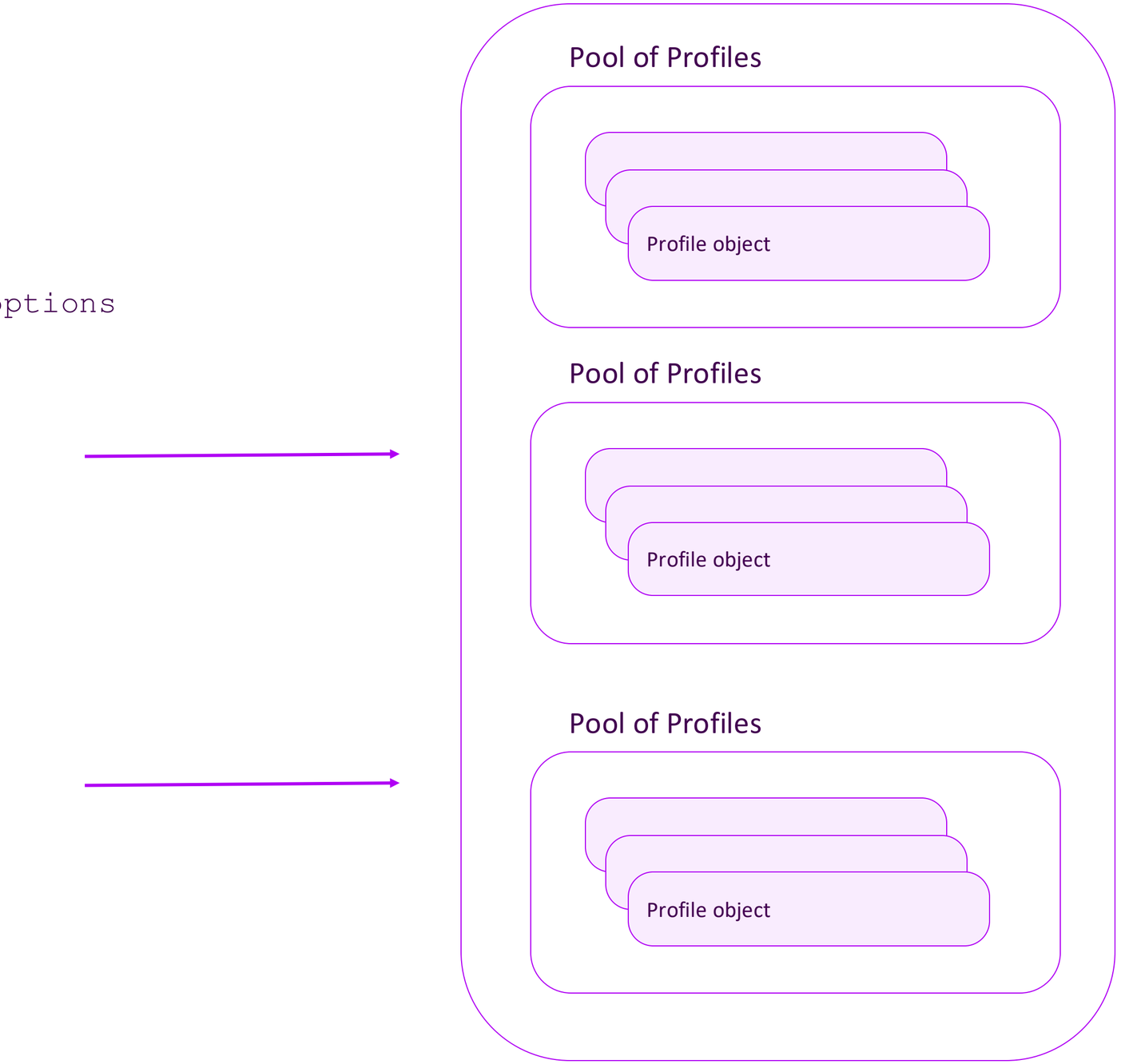
Object Pooling

```
POST /api/v3/navigation/search-travel-options
```

Give me a set of Profiles I can use in my algorithm

Navigate

Done, release my pool and reset



Performance gain of over 10%!
A lot less work for the GC

Improving Performance by reducing allocations

Improving Memory Usage by reducing object sizes

Connection Object

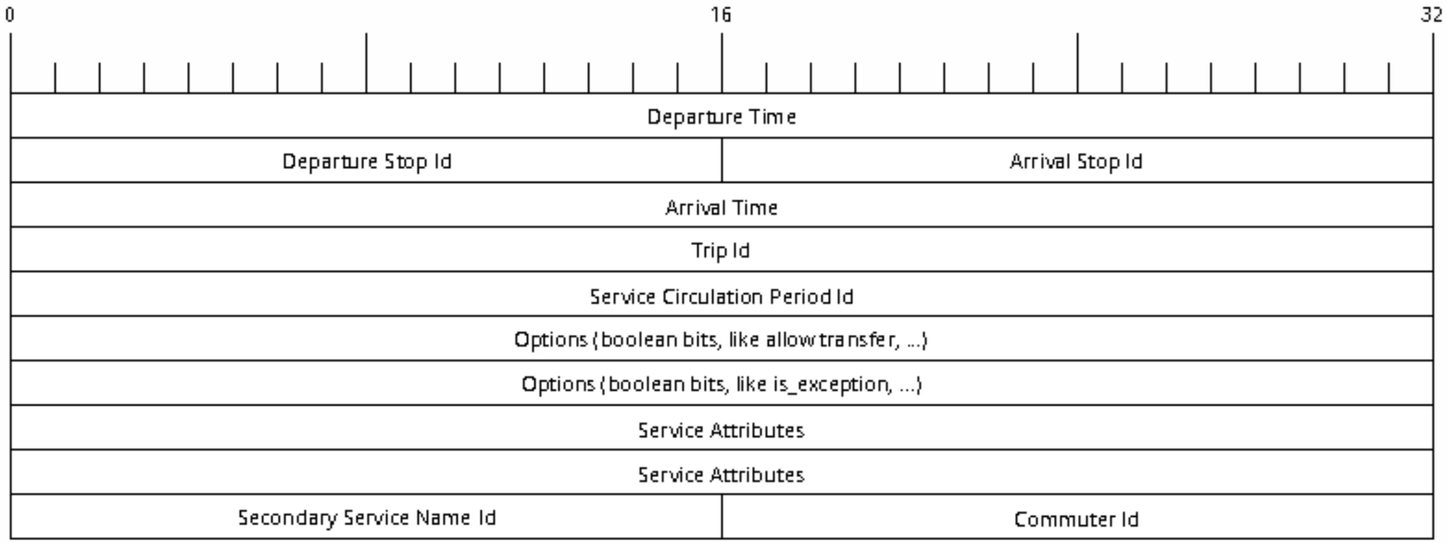
departureStation	1
departureTime	1726479782
arrivalStation	3
arrivalTime	1726479782
allowBoarding	☑ true
allowTransfer	☑ true
allowDisembarking	☑ true

```
16 (overhead) + (depStation) 4 + (depTime) 8
+ (arrStation) 4 + (arrTime) 8
+ 3 (3x boolean)
+ 1 alignment
= 44 bytes

times 50m = ~2.2GB connections
```

Int = 4 bytes
Long = 8 bytes
Boolean = 1 byte
Object overhead = 16 bytes

Map to IntArray



Int 0 contains the departure time
Int 1 first 16 bits contains departure station
Int 1 second 16 bits contains arrival station
etc

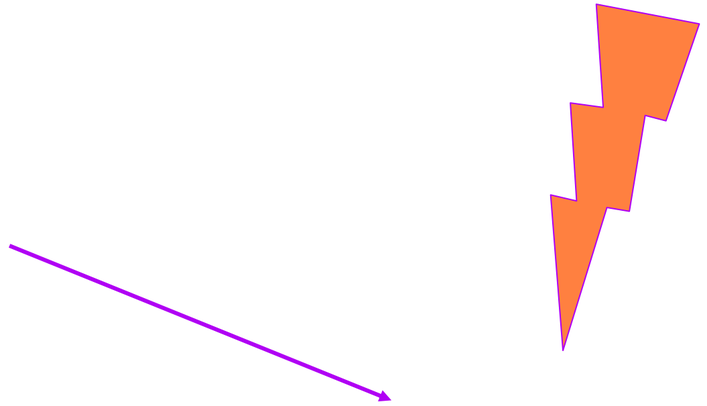
1 GB left! reduction of 50%

Before

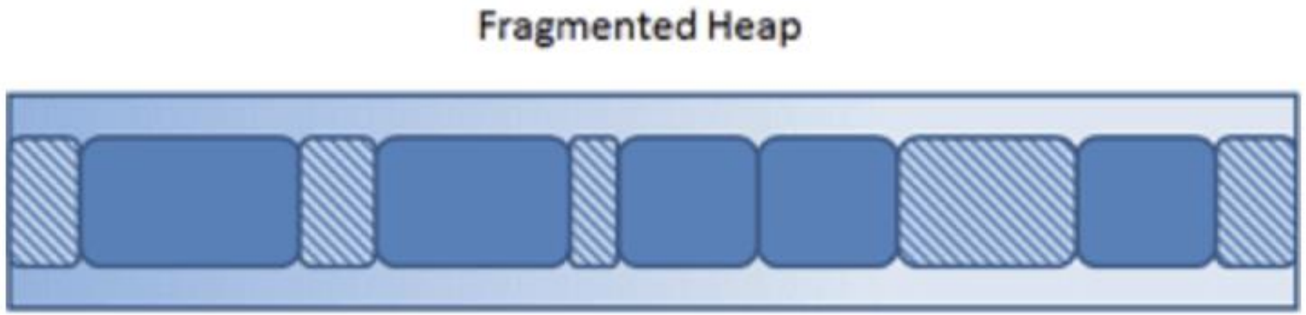
```
List<Connection>
50m x 4 bytes = 200M
+ 50m x 44 bytes = 2.2GB (separate objects)
```

After

```
Every connection is now 4 Ints
50m * 4*4 bytes = 800M (one
object)
```



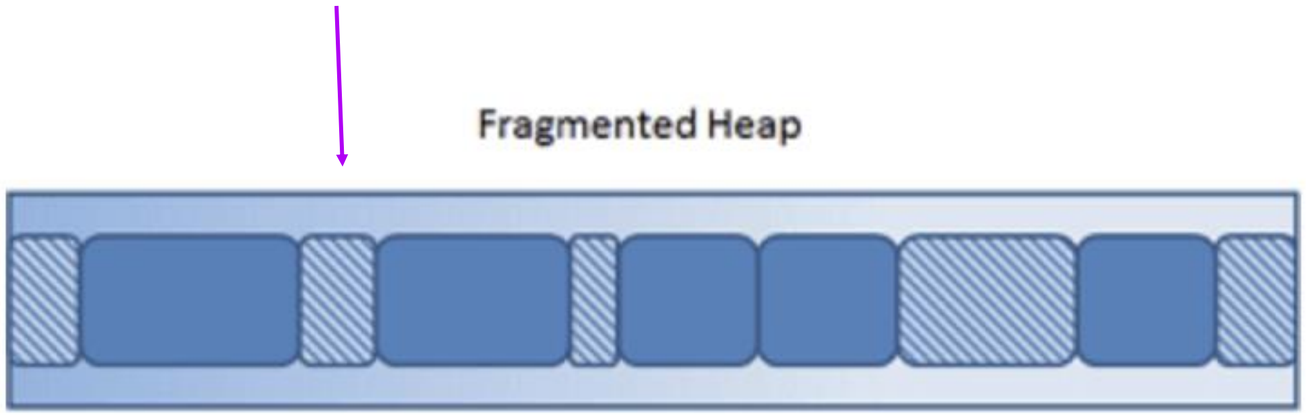
Out of Memory!



After after

```
val blocks = ArrayList<IntArray>() // huge connection list
val conn = blocks[i / 1000000][i % 1000000]
```

4MB Block fits in gap



Improving Performance by reducing allocations

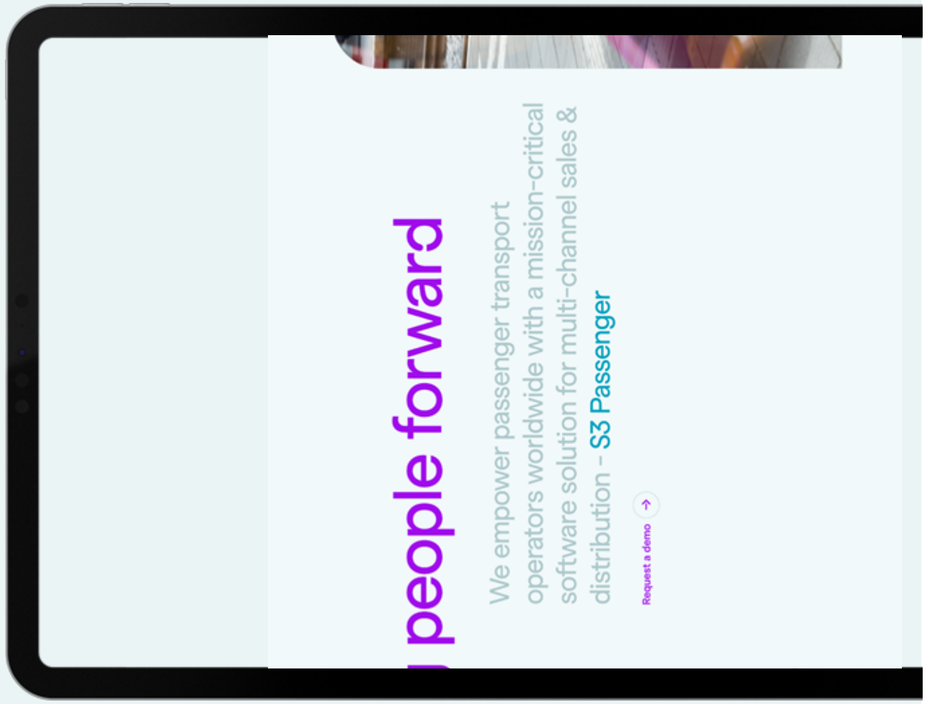
Improving Memory Usage by reducing object sizes



Questions?

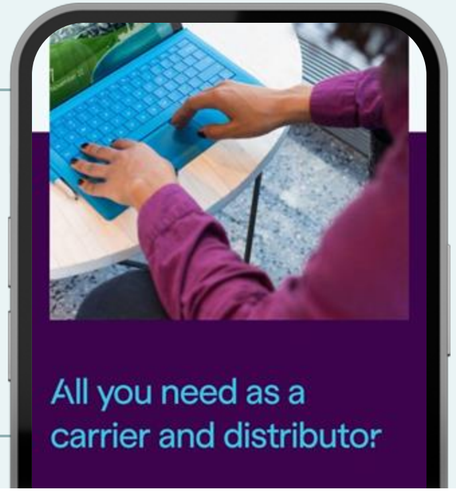
Thank you

for your attention



For more information about S3 Passenger, the implementation approach and our knowledge hub, go check out

sqills.com



Enjoy the
food and drinks

Up next: Filip Ivanov
How to use CSS Flexbox to implement creative designs

Feedback?
Share it with us!



Weird flex, but ok – a bit about flexbox layouts

September 20, 2024

Filip Ivanov

EL NIÑO

Table of contents

- Flexbox - a brief history.
- Flexbox fundamentals.
- Flexbox properties.
- Further reading.

Flexbox – a brief history

a.k.a how did we end up here?

The web used to be (more) simple...

The web used to be (more) simple...



The web used to be (more) simple...



- Common resolutions
- Common aspect ratios
- Consistent and simple design

2.5 Million Titles

Text Only

SEARCH BY
Author, Title, Subject
Keyword
ISBN
Advanced Query

BUY BOOKS
Shopping Cart
Checkout

SUPER ROOM
Computer & Internet

MILES OF AISLES
Business
Science Fiction
Nonfiction
Literary Fiction
Romance
Children's Books
And Many More . . .

LISTS
Titles in the News
Bestsellers
Award Winners
Amazon.com 500
Hot 20
Mystery 50
Literary 50
Computer 50
Business 50
Science Fiction 50

ARTICLES & INTERVIEWS
Amazon.com Journal
Eclectica
In Their Own Words

SIGN-UP
Eyes
Expert Editors

BIG FUN
Win Prizes
Wanna Play?
Customer Reviews

PARTNERS
Associates
Publishers
Authors

Welcome to Earth's Biggest Bookstore

Amazon.com

1.5 Million Books in Print & 1 Million Out-of-Print Books

40% off the Amazon.com 500! ~ Browse our Miles of Aisles
Win $1,000 of books ~ Earn money selling books as an Amazon.com Associate!

First-Time Visitors Please Click Here

May 28th--New on Our Shelves



Joseph McBride's new biography gives a fine spiel on *Steven Spielberg*. *The Dilbert Future* has arrived in our Computer & Internet Super Room. Plus, new titles on our Literary Fiction, History, and Nonfiction shelves **discounted up to 40%**.

Book of the Day

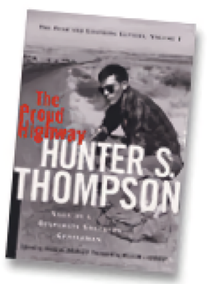
Classic BOTD
Undiscovered BOTD
Science Fiction BOTD
Nonfiction BOTD
Mystery BOTD
Wacky BOTD

6 new books every day for the next 1,000 years

amazon.com journal

Amazon.com Journal: *The Proud Highway* peeks into Hunter S. Thompson's postal past during the Fear and Loathing years... *Marcus & Diana* is a romp--pure, if not simple... Funny and illuminating, former U.S. Labor Secretary Robert Reich's *Locked in the Cabinet* proves even a friend of Bill can have trouble capturing the ear of Bill... An ill-fated expedition up Mount Everest makes for can't-put-it-down reading in Jon Krakauer's *Into Thin Air*.

And in Eclectica: Dertouzos redux. This, and more, in *Amazon.com Journal*, where titles are offered at **up to 40% off list price.**





Titles in the News

The *New York Times Book Review* books are in! If you ask us, Arundhati Roy's novel, *The God of Small Things*, is a big thing indeed... *Che* is a stunningly good book about the man who drove Cuba's revolution but found disappointment in its success... And *The Perfect Storm* is a dramatic, gale-by-gale account of the 1991 "storm of the century" that pummeled the northeastern United States.

Check out the latest books reviewed in *The New York*

The web got complicted...

The web got complicted...



The web got complicted...





The web got complicted...







The web got complicted...









How do we handle this complexity?

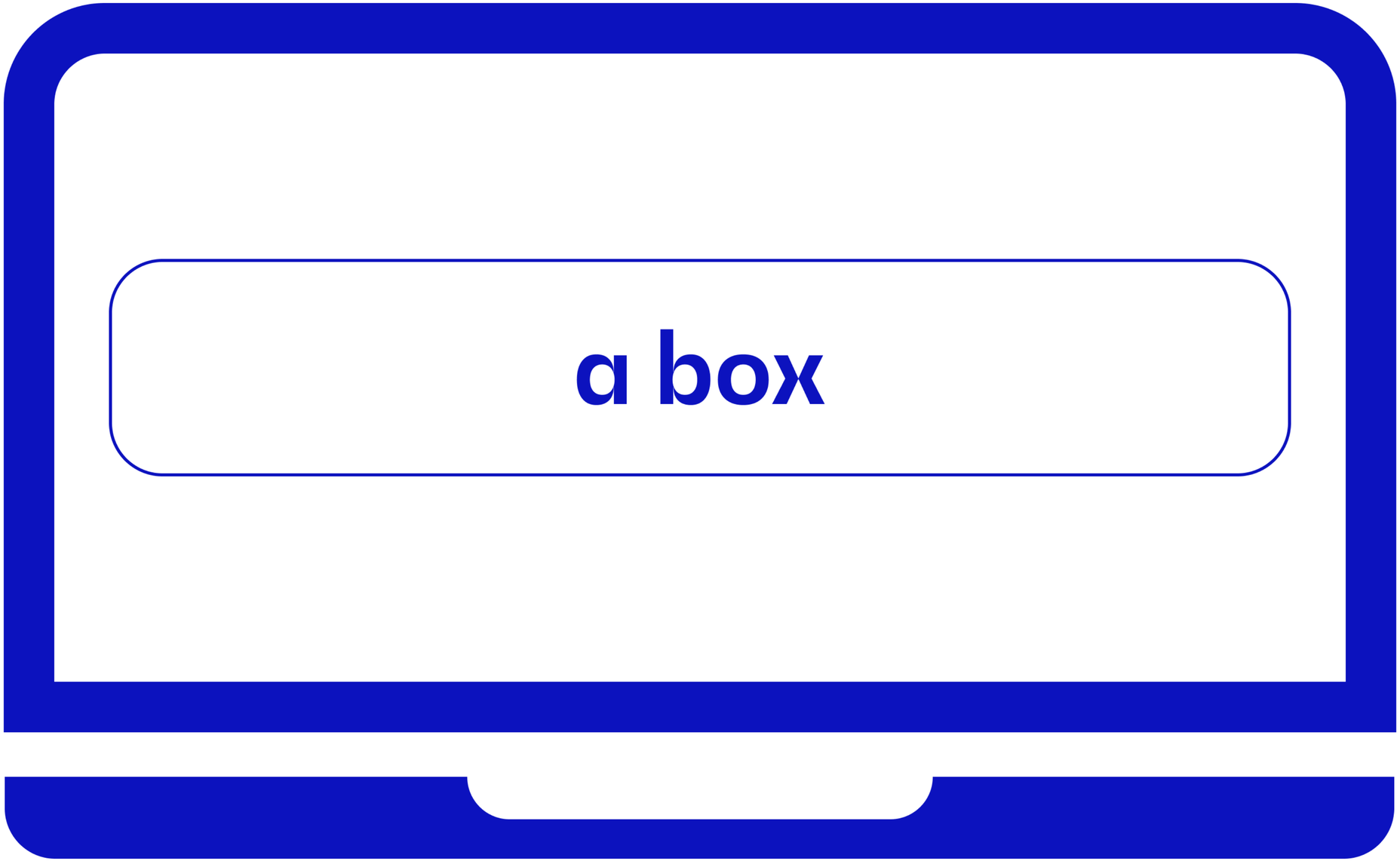
Flexbox components

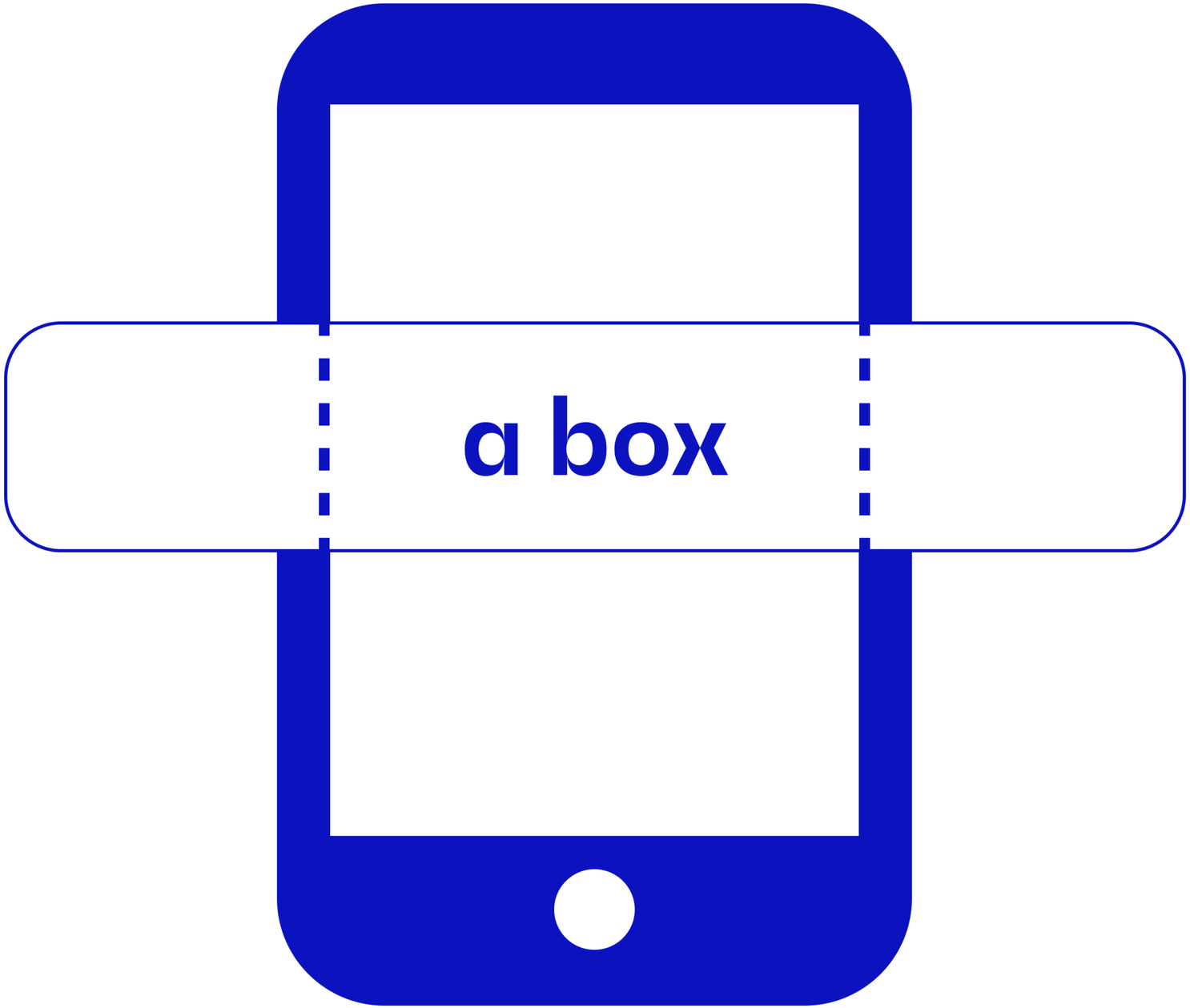
and general responsive layouts

Flexbox fundamentals

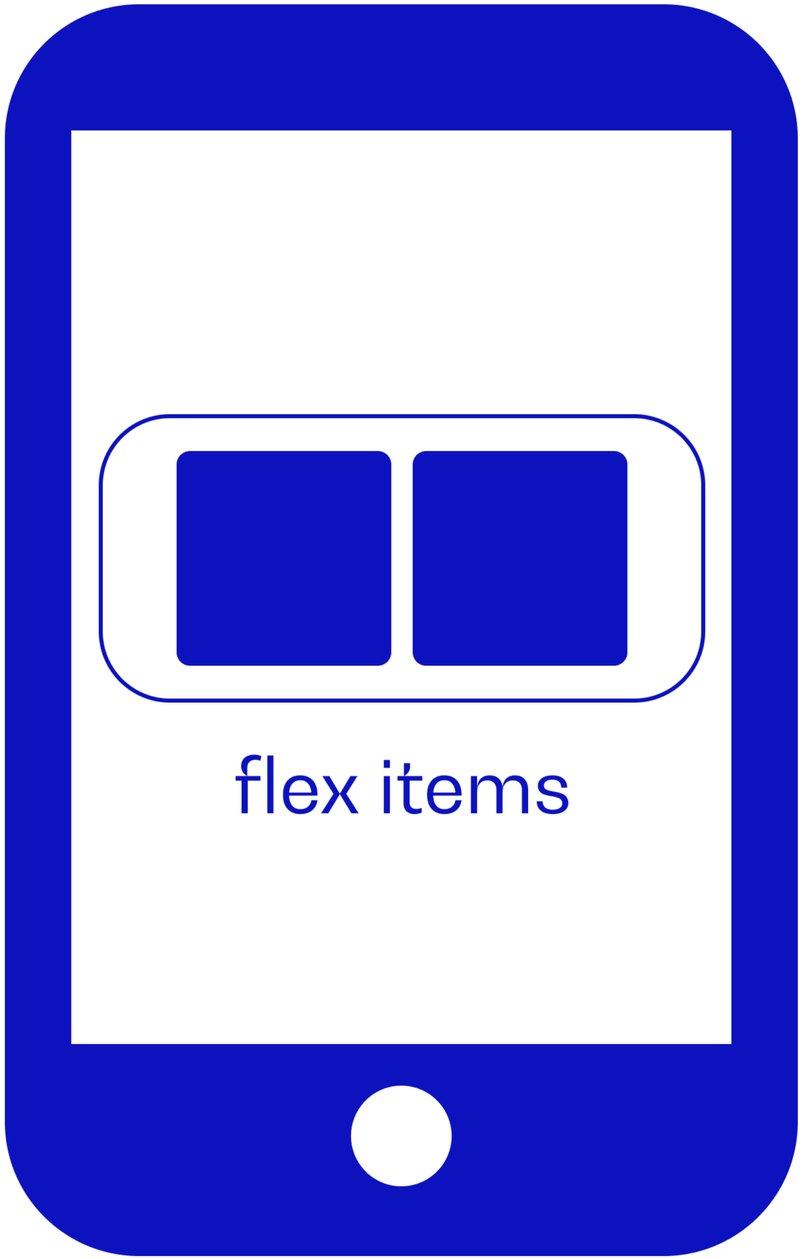
a box

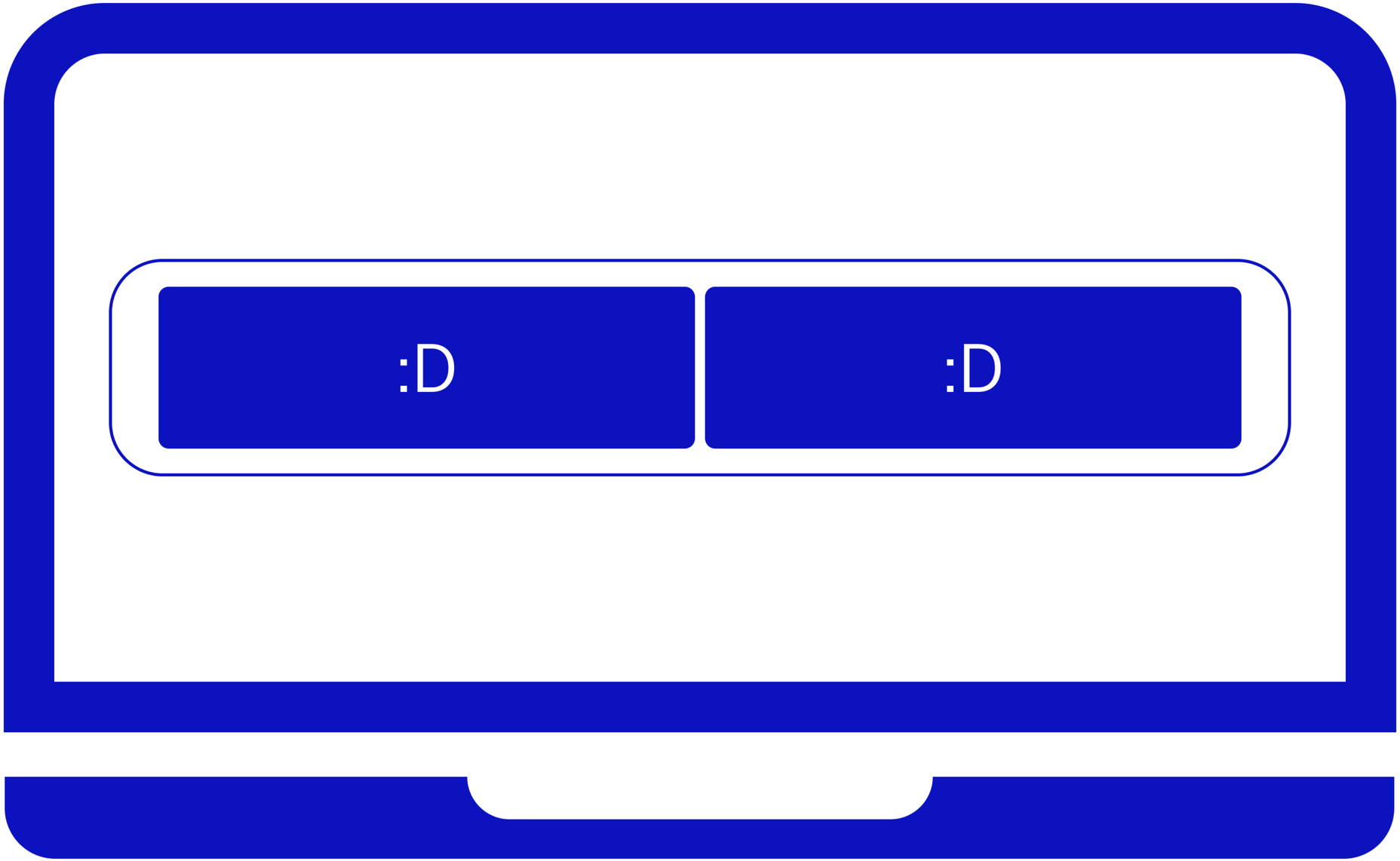
container

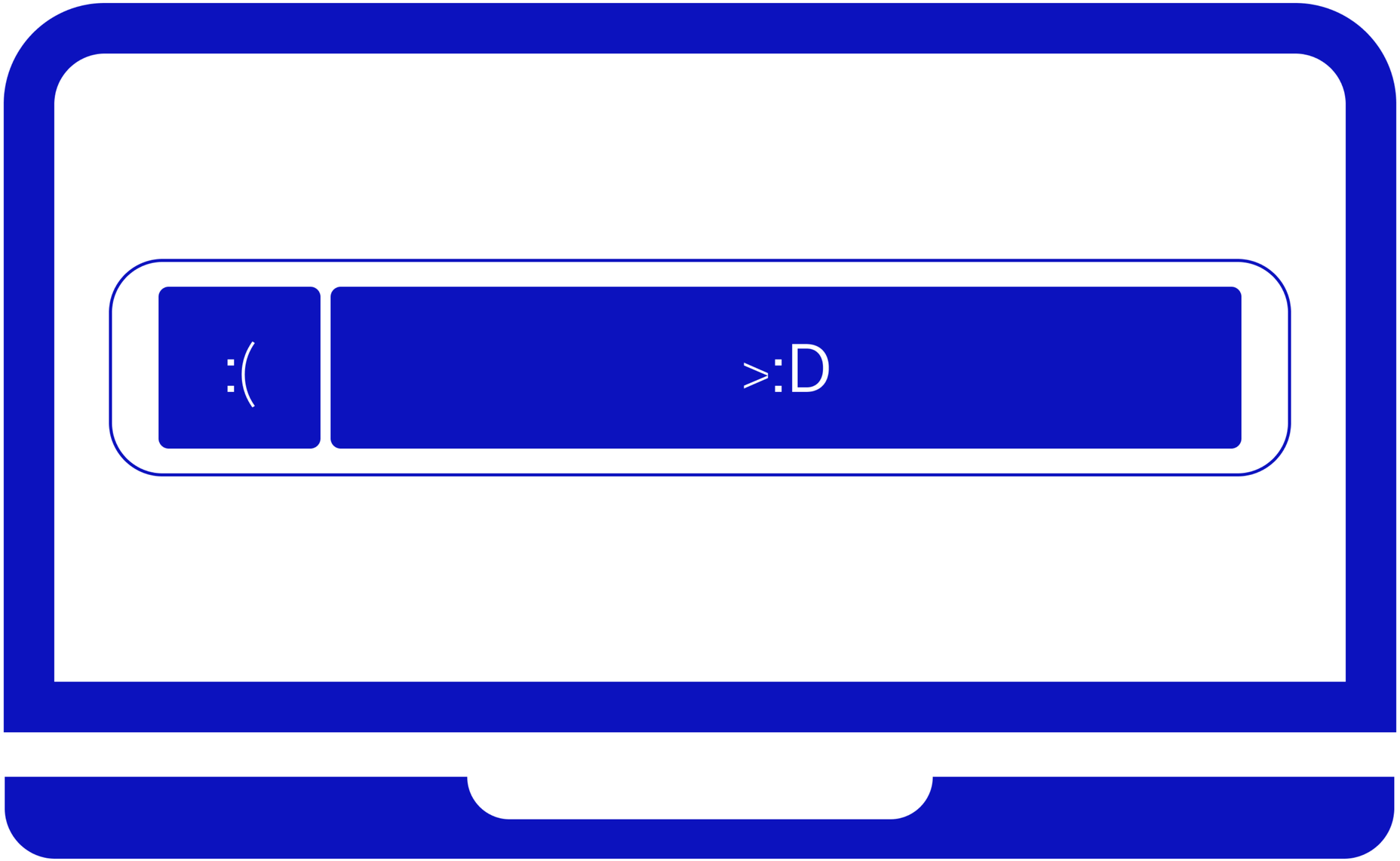

a box

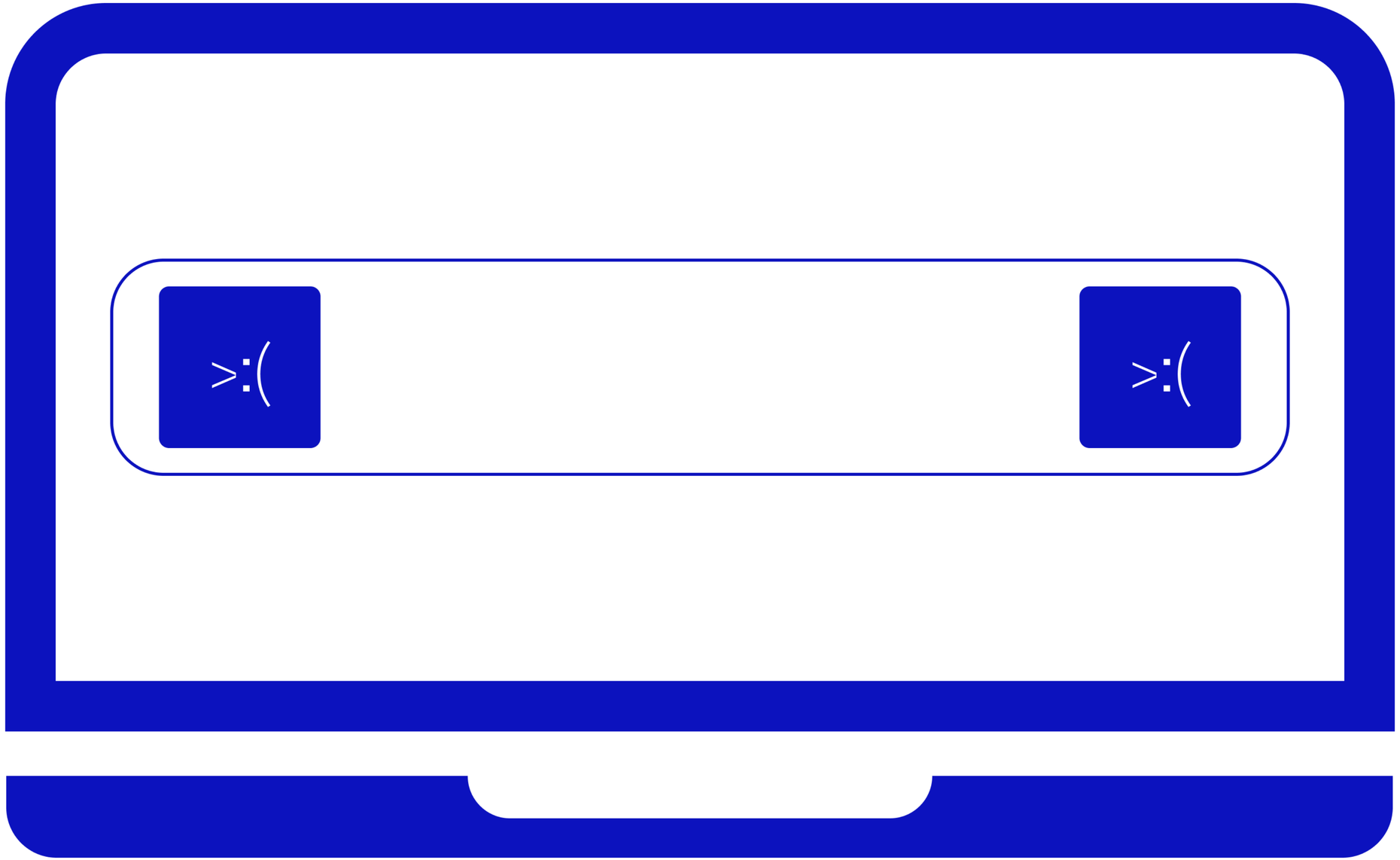

a box


a flexbox
flex container


flex items


:D
:D


:(
>:D



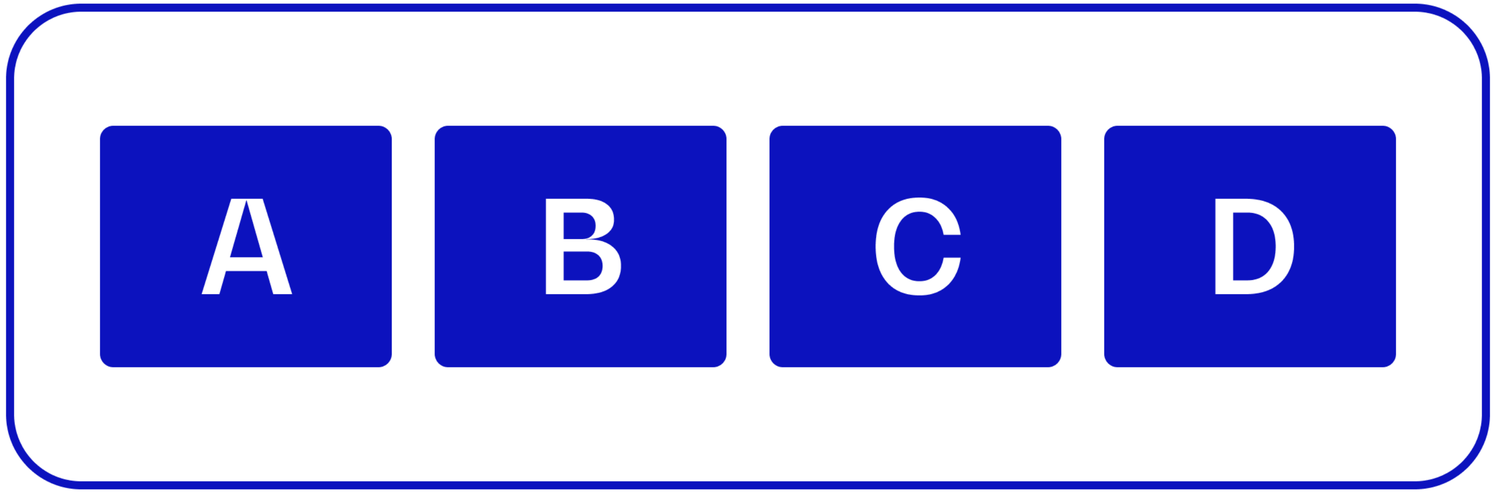
Flexbox properties

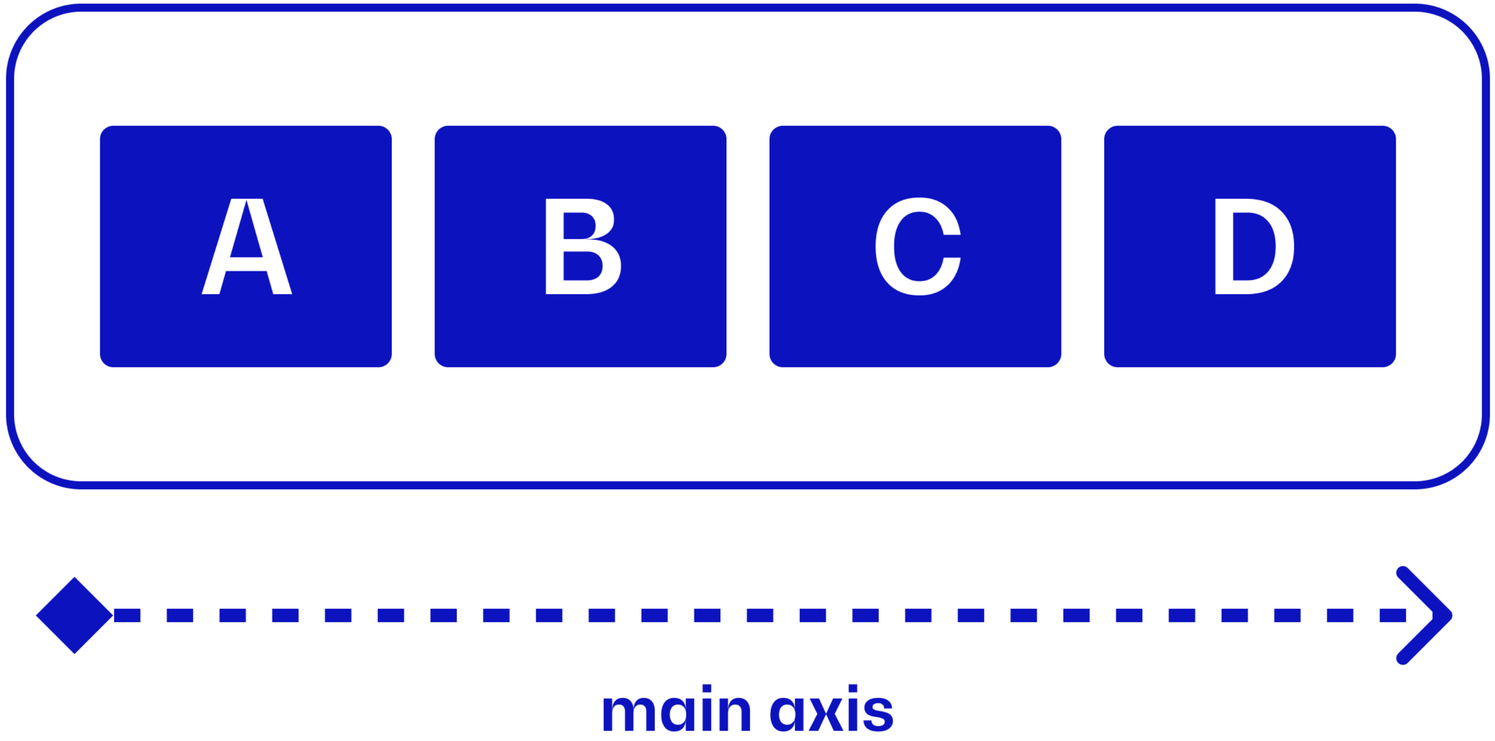
- Flex container (parent)
- Flex items (children)

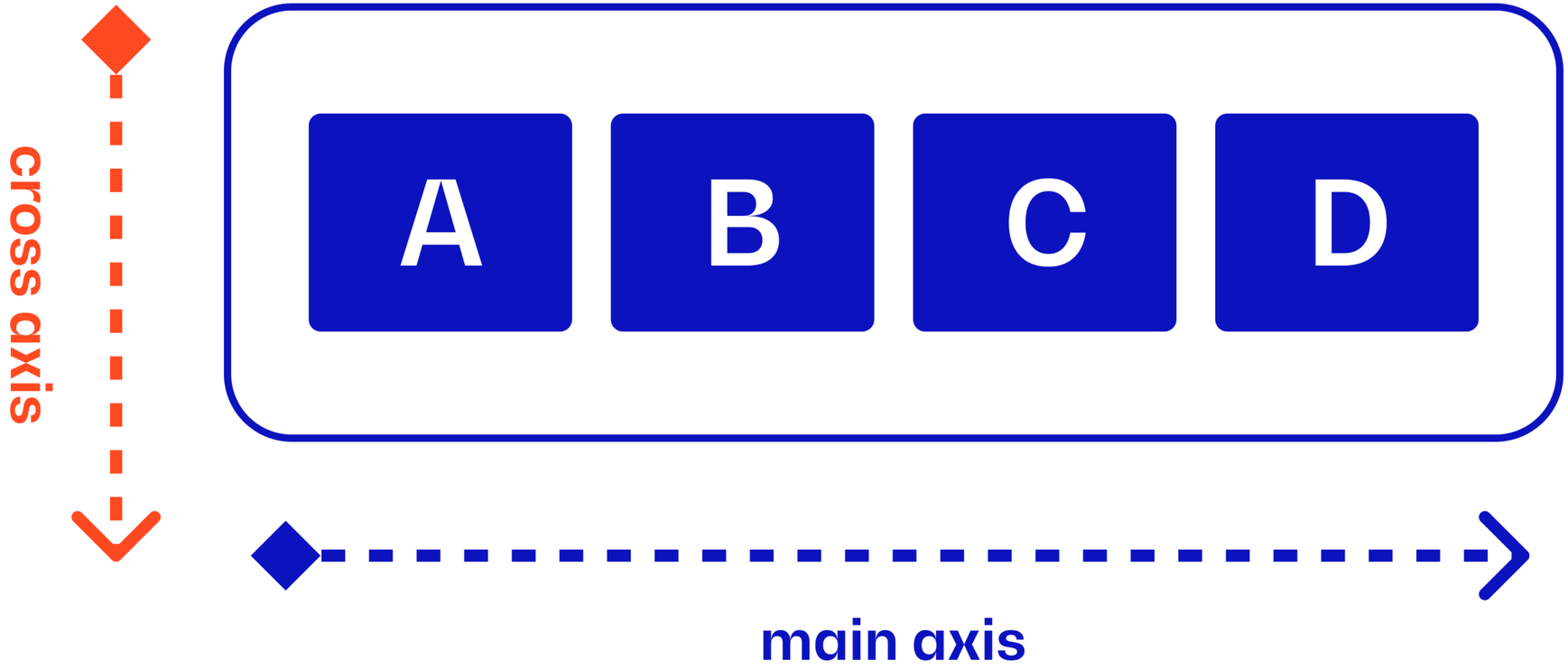
boring container :(

```
.boring-container {
    display: flex;
}
```

flexbox container :D

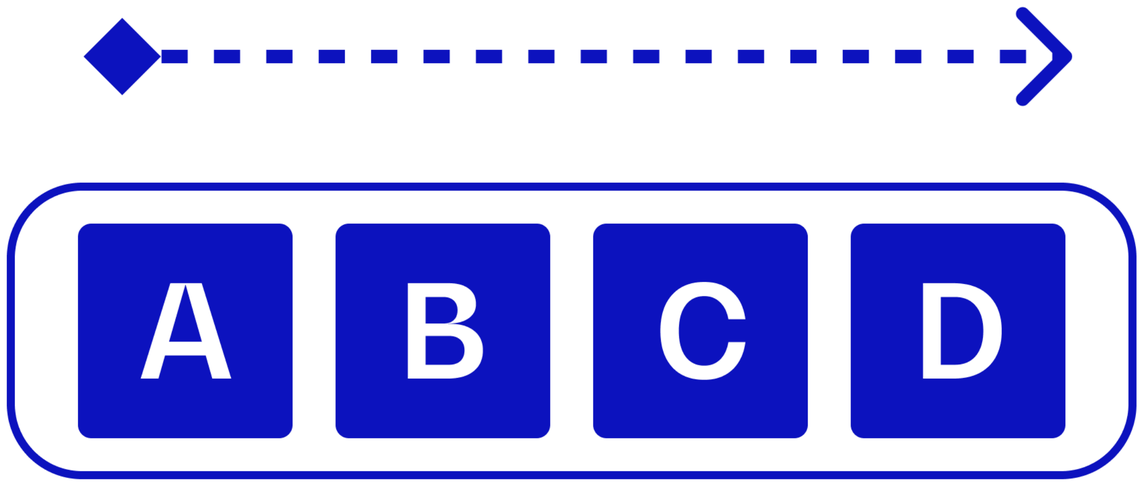

A
B
C
D


A
B
C
D
main axis


A
B
C
D
cross axis
main axis

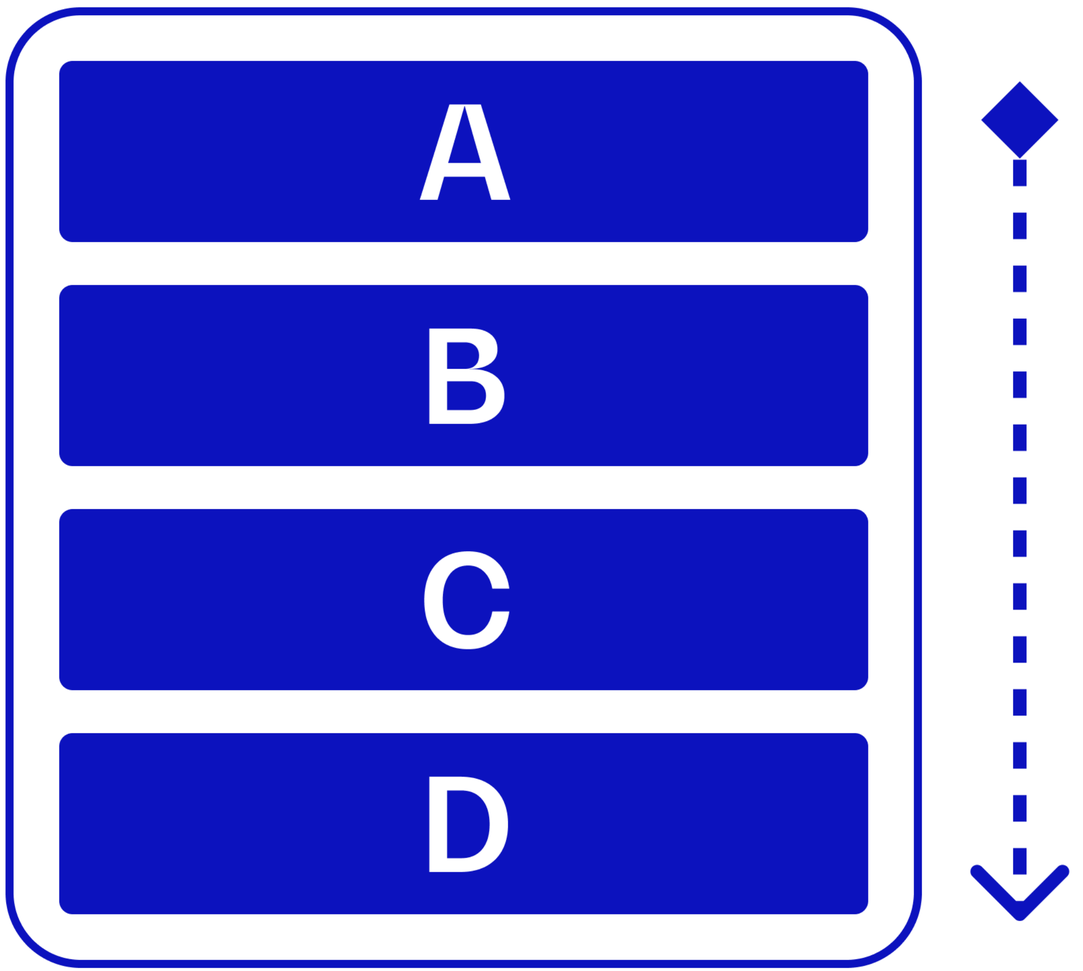
Flex direction

```
.boring-container {
    flex-direction: row;
}
```



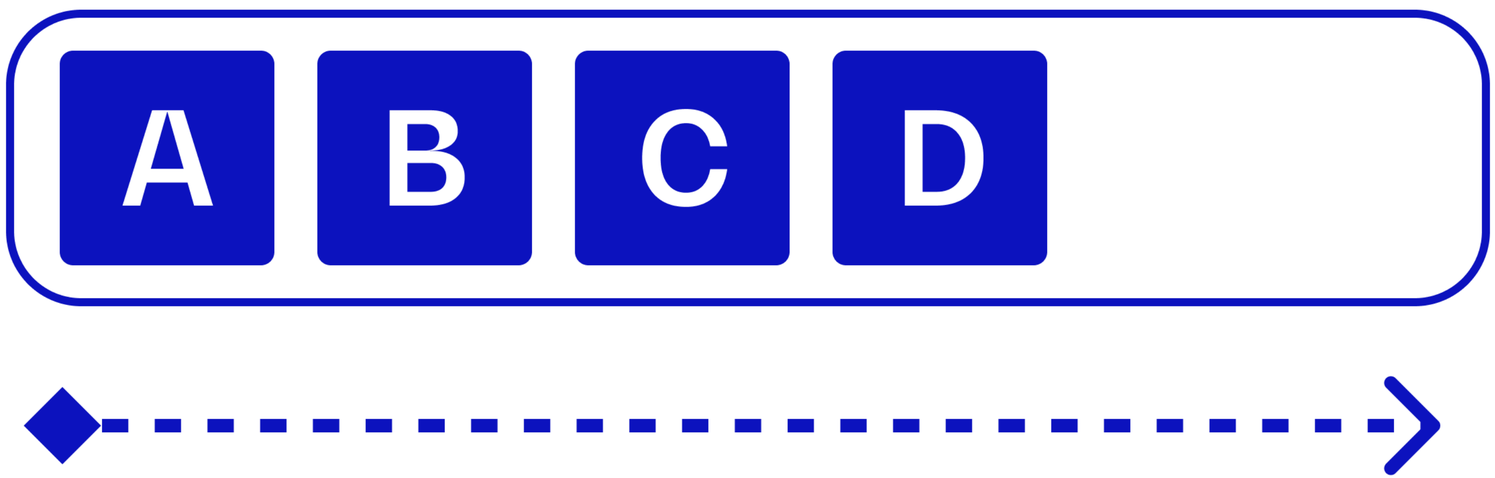
Flex direction

```
.boring-container {
    flex-direction: column;
}
```



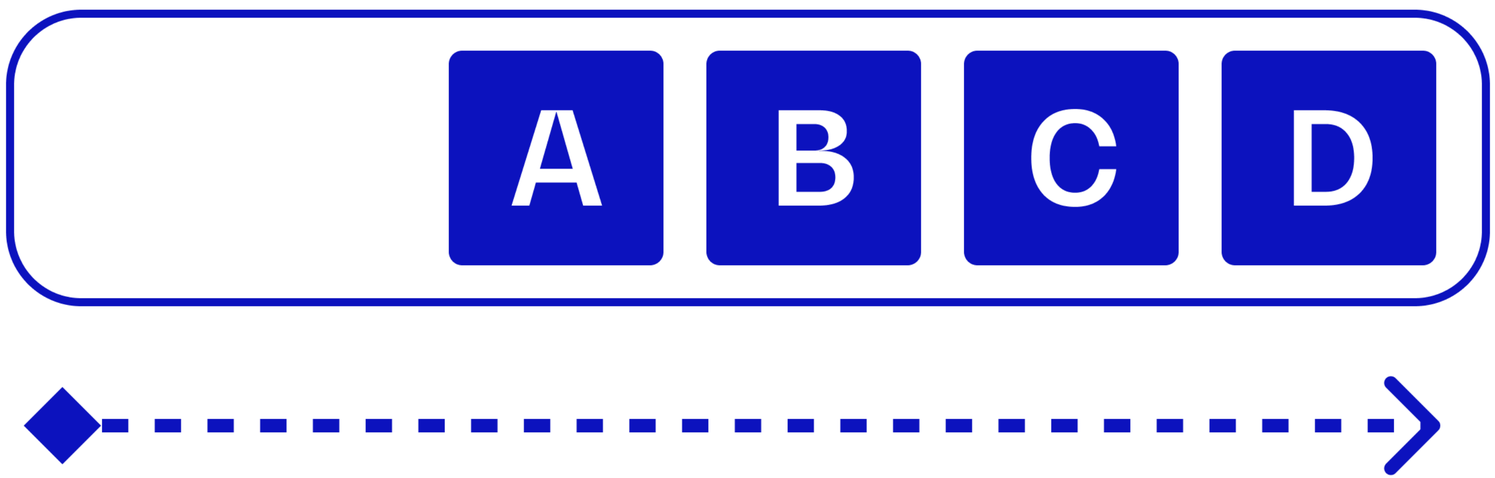
Justify content

```
.boring-container {
    justify-content: flex-start;
}
```



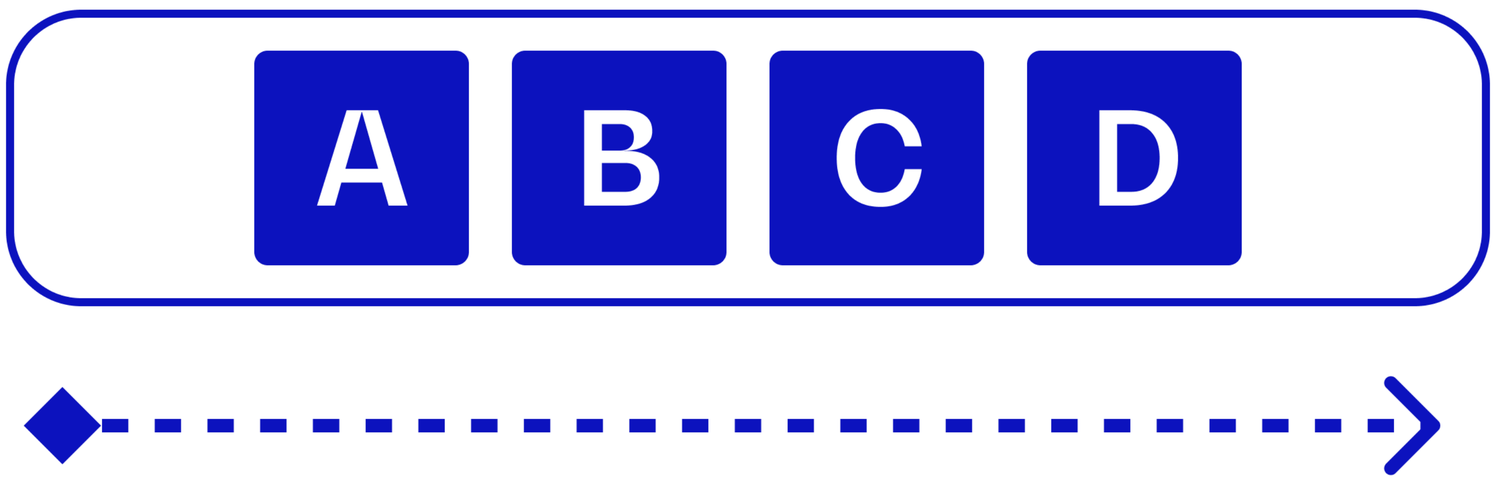
Justify content

```
.boring-container {
    justify-content: flex-end;
}
```



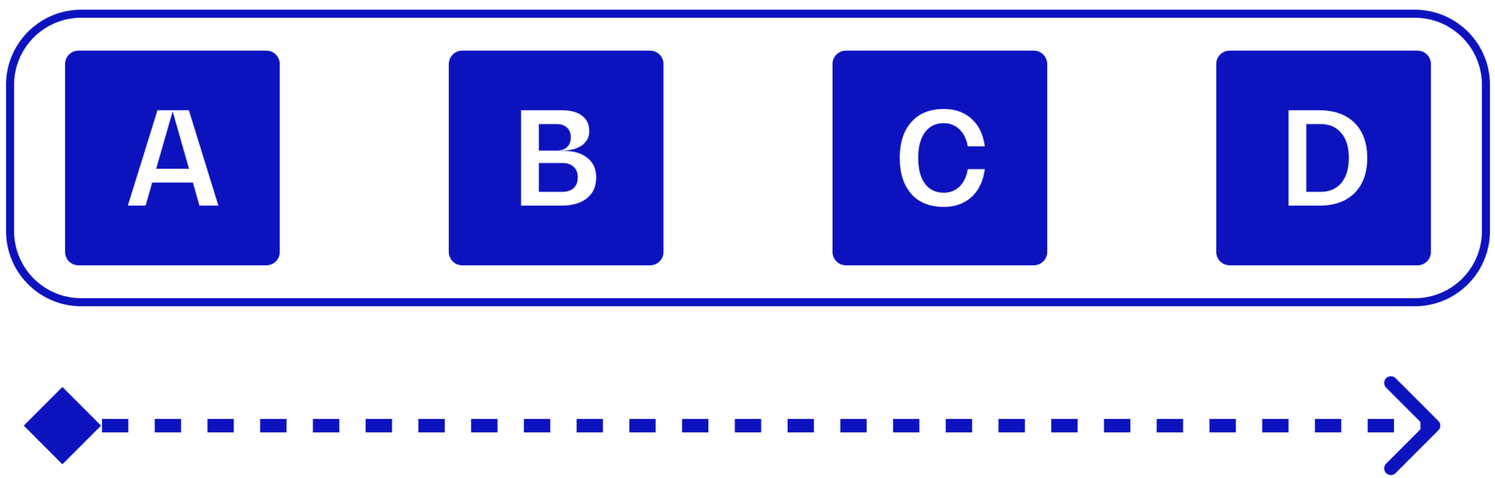
Justify content

```
.boring-container {
    justify-content: center;
}
```



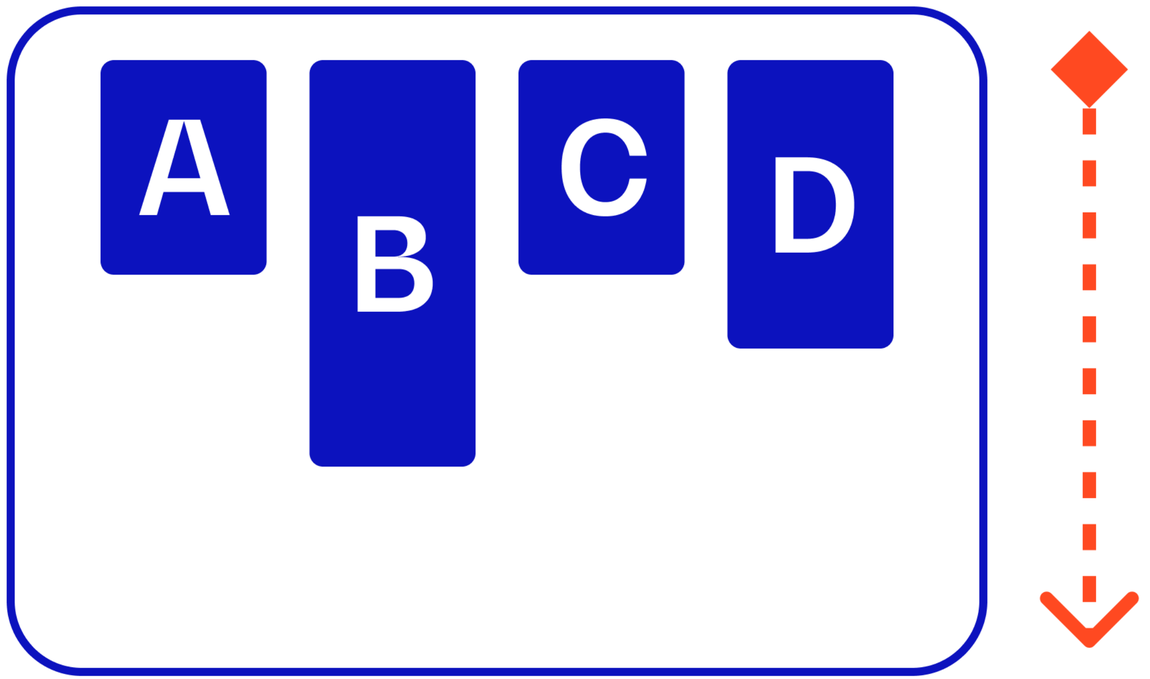
Justify content

```
.boring-container {
    justify-content: space-between;
}
```



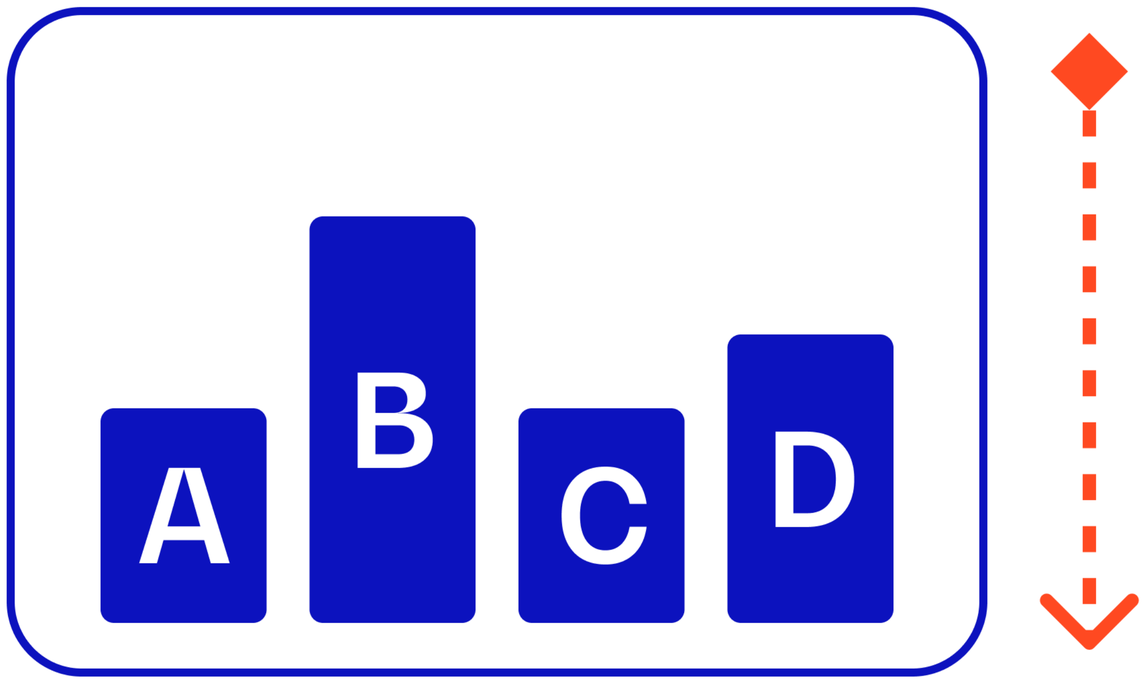
Align content

```
.boring-container {
    align-content: flex-start;
}
```



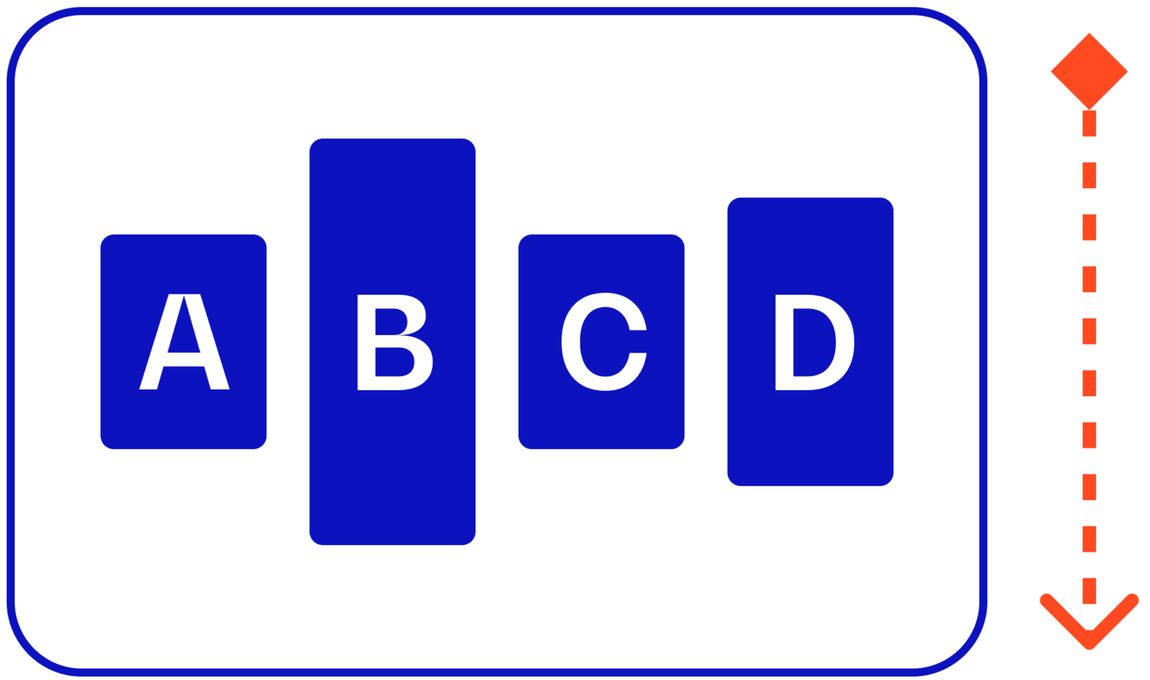
Align content

```
.boring-container {
    align-content: flex-end;
}
```



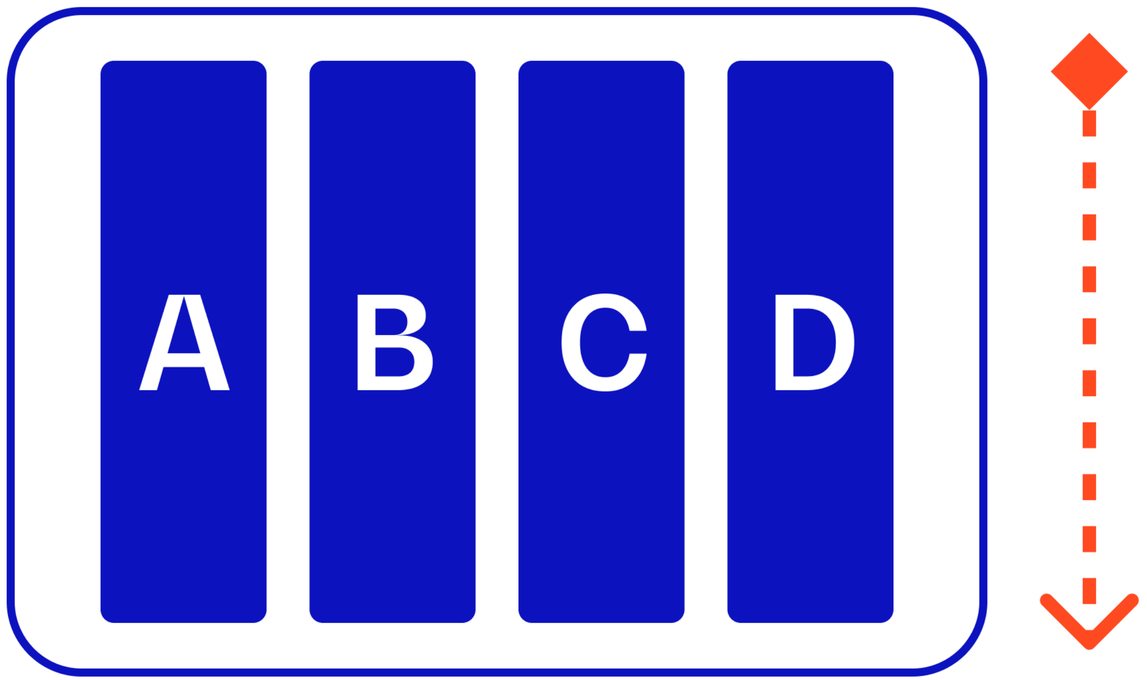
Align content

```
.boring-container {
    align-content: center;
}
```



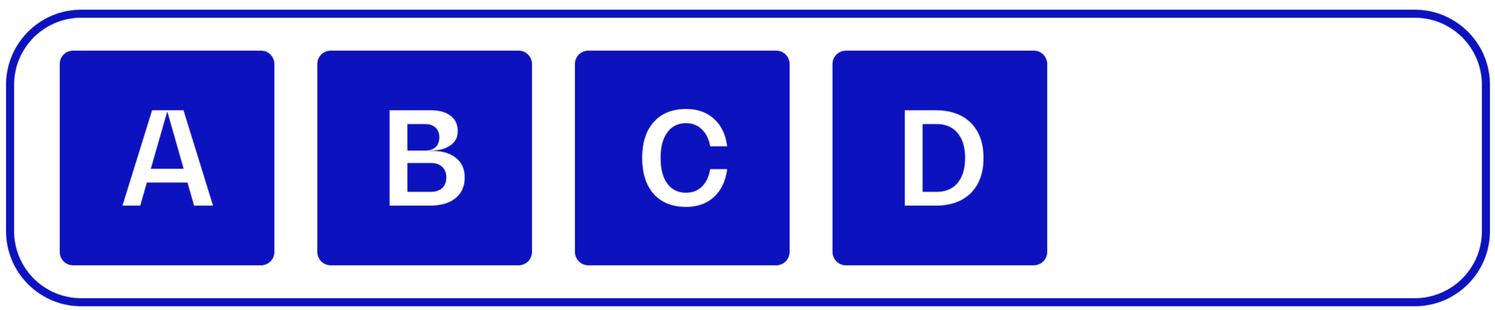
Align content

```
.boring-container {
    align-content: stretch;
}
```



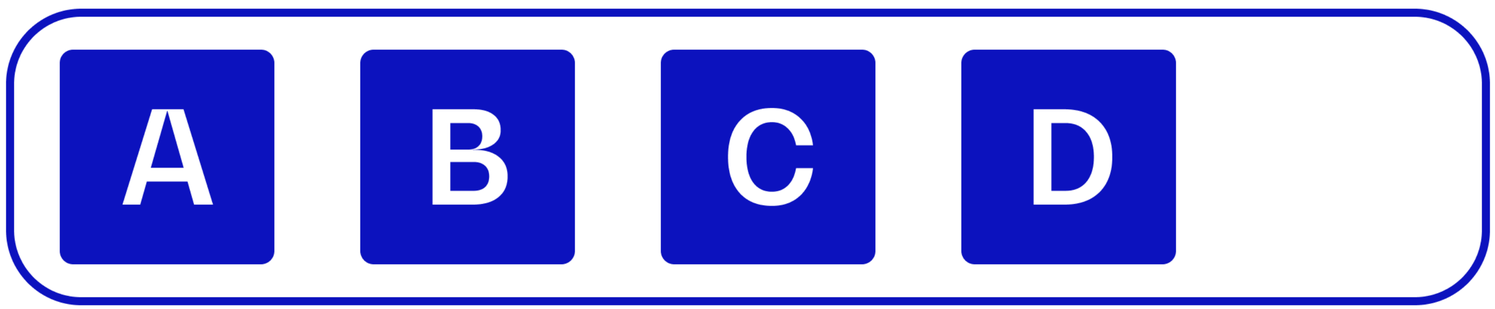
Gap

```
.boring-container {
    gap: 32px;
}
```



Gap

```
.boring-container {
    gap: 64px;
}
```



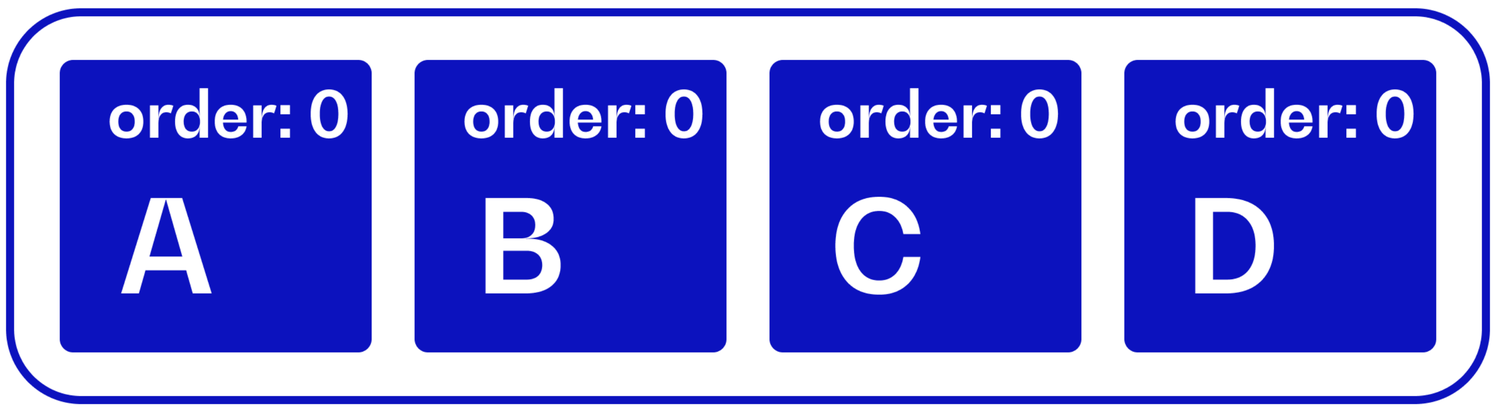
Gap

```
.boring-container {
    gap: 16px;
}
```



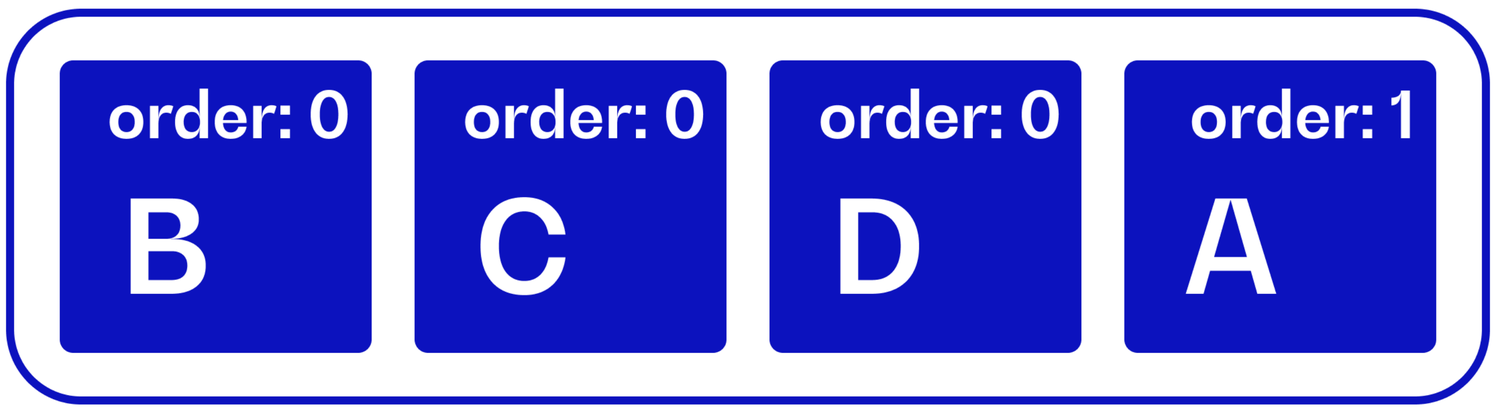
Flex item properties

Order



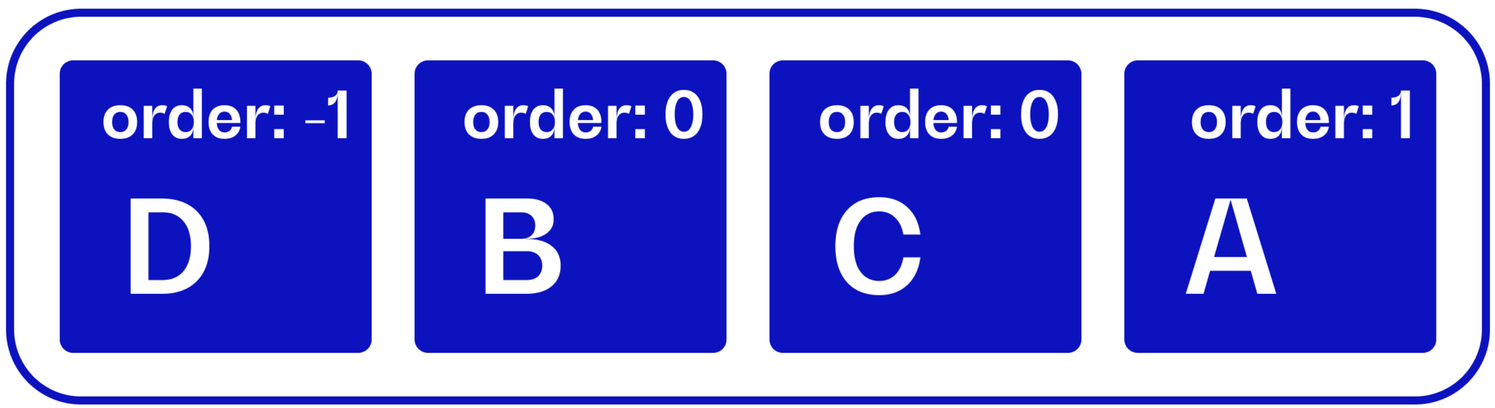
Order

```
.item-a {
    order: 1;
}
```

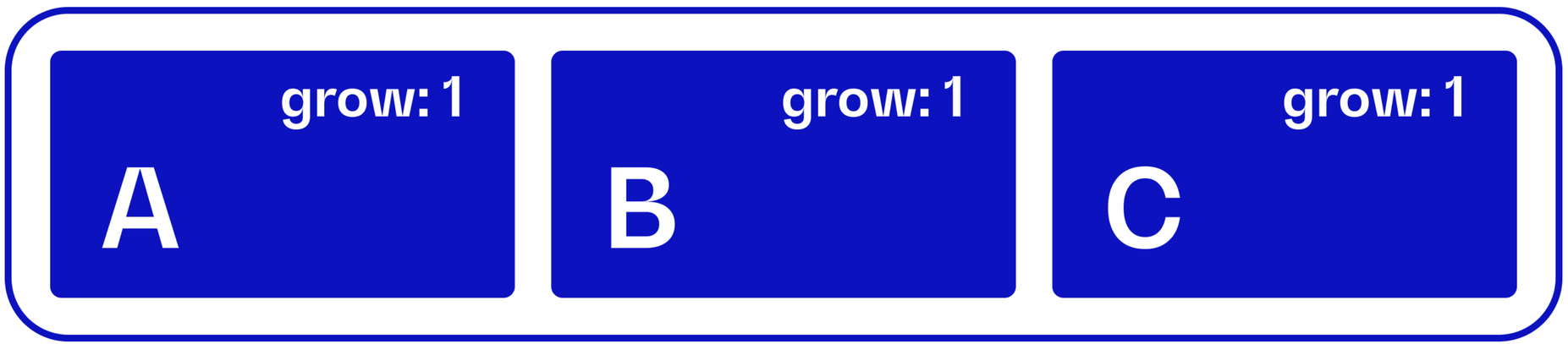


Order

```
.item-d {
    order: -1;
}
```



Flex grow



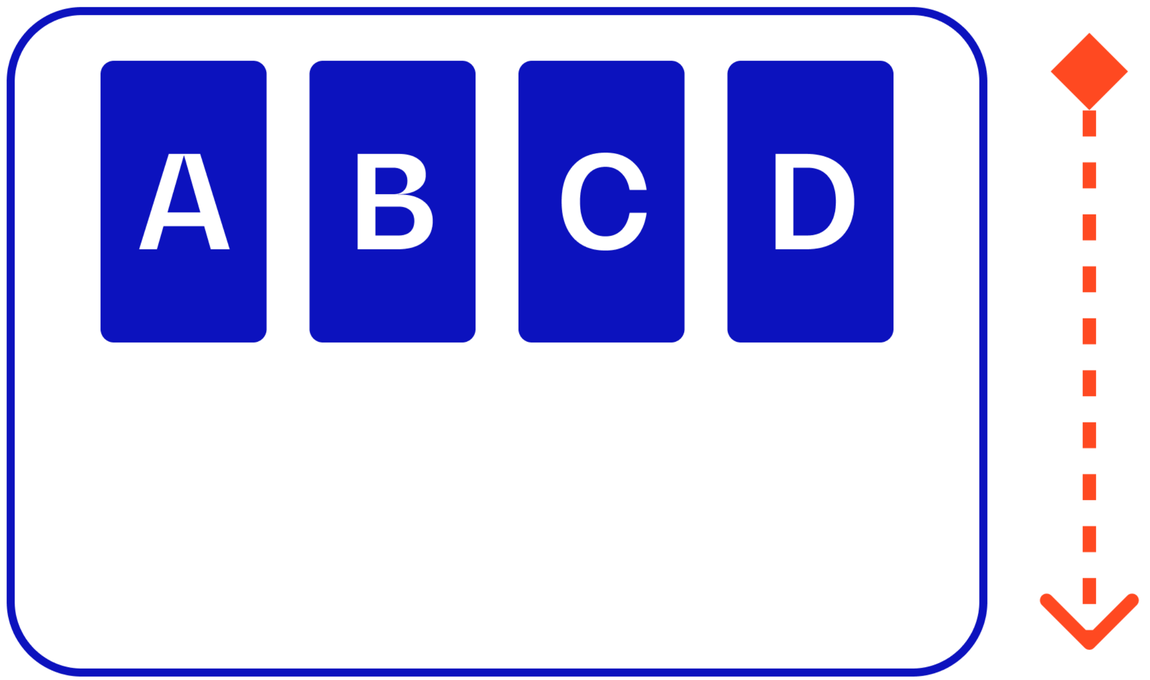
Flex grow

```
.item-b {
    flex-grow: 2;
}
```



Align self

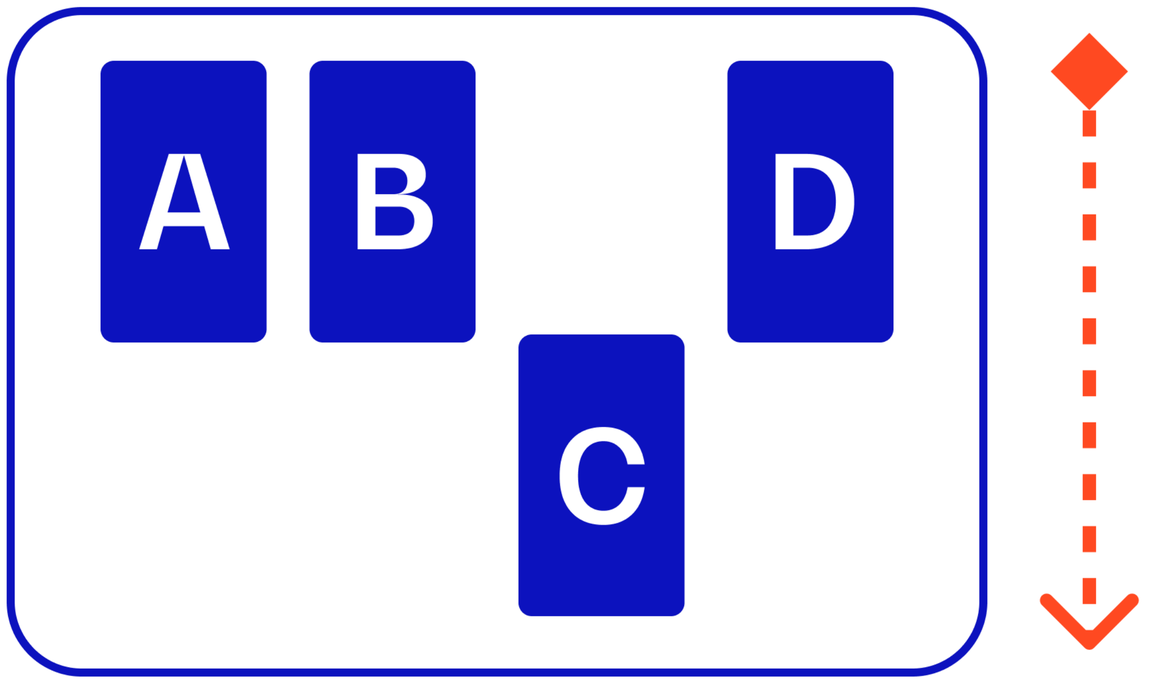
```
.boring-container {
    align-content: flex-start;
}
```



Align self

```
.boring-container {
    align-content: flex-start;
}

.item-c {
    align-self: flex-end;
}
```



Conclusion

There’s much more to know, but don’t worry about knowing it all :D

Further reading

- **More in-depth about flexbox**
  CSS Tricks - *https://css-tricks.com/snippets/css/a-guide-to-flexbox/*

- **More about the history of website layouts**
  Juxtopposed - *https://www.youtube.com/watch?v=ckEdTprbQas*

Questions/ remarks?

Filip Ivanov
filip@elnino.tech

EL NIÑO

Feedback?
Share it with us!



Enjoy the
food and drinks

Up next: Joris Blaak

How to Run the Leading Indoor Rowing Championship: A Tech Deep Dive

MoveLab

How to Run the Leading Indoor Rowing Championship



Today

- About indoor rowing
- The racing itself
- Orchestration
- UI challenge

A rowing machine



A better rowing machine



This is what we’re talking about



The event

- 60+ rowing machines with tablets
- 2500 participants in a single day
- Races of 2000m

Our job

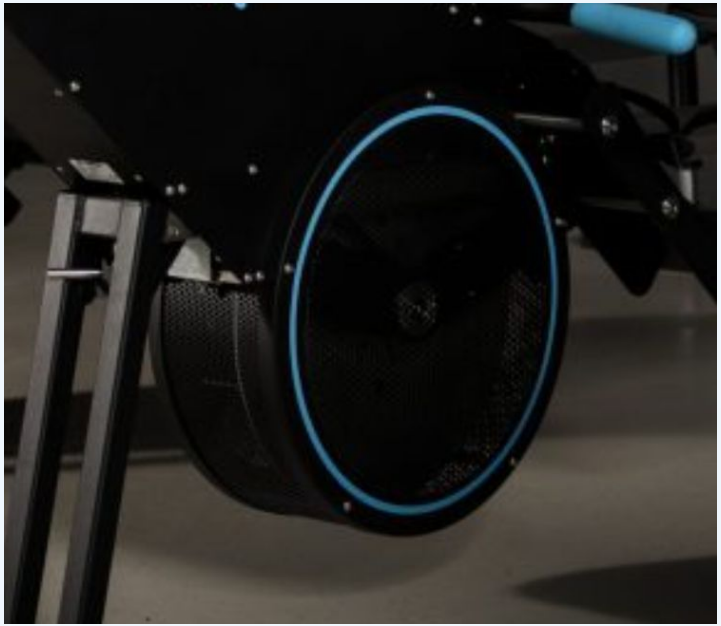
- Read sensor pulses from the flywheels
- Find the person who yanks the chain the hardest

Our job

- Assign participants to right lanes
- Trigger start sequence
- Read sensor pulses from the flywheels
- Convert to something sensible
- Share statistics with millisecond precision
- Stream realtime racing data to the media
- Display synchronised realtime racing data
- Determine guaranteed finish times

The racing itself

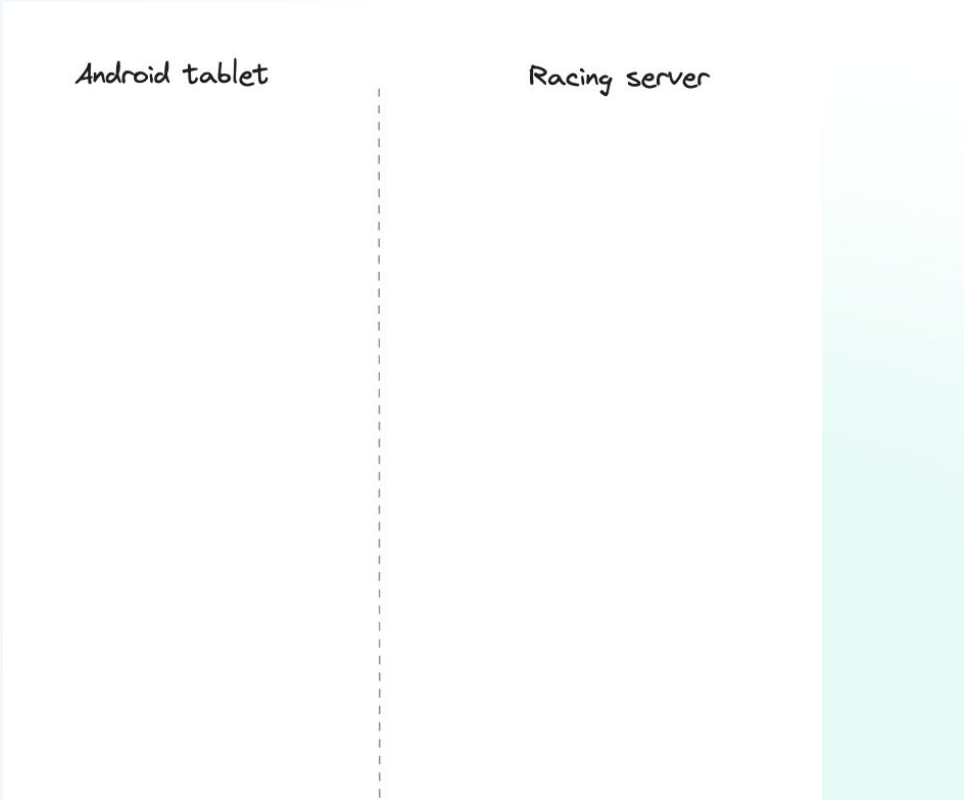
- FYI this is where we start





The racing itself

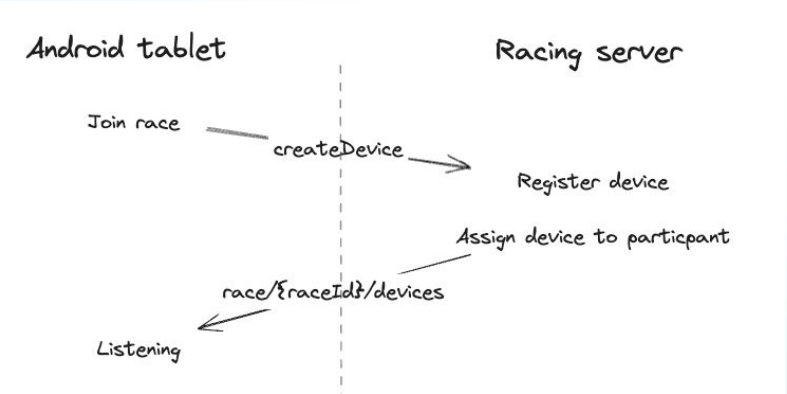
- Many tablets and a single server
- Server technology
  - Node.js
  - Express
  - Apollo (GraphQL)
  - Socket.IO
  - RxJS
  - TypeScript



The racing itself

- Announce as a device (tablet red.)

```
mutation {
   createDevice(
      raceId: "the-id",
      participantId: "some-id"
      ) {
        id
        raceId
        participantId
        isWheelSpinning
        ping
        lastPingReceivedMillis
   }
}
```



- Start listening on events

```
`race/${raceId}/devices`
```

The racing itself

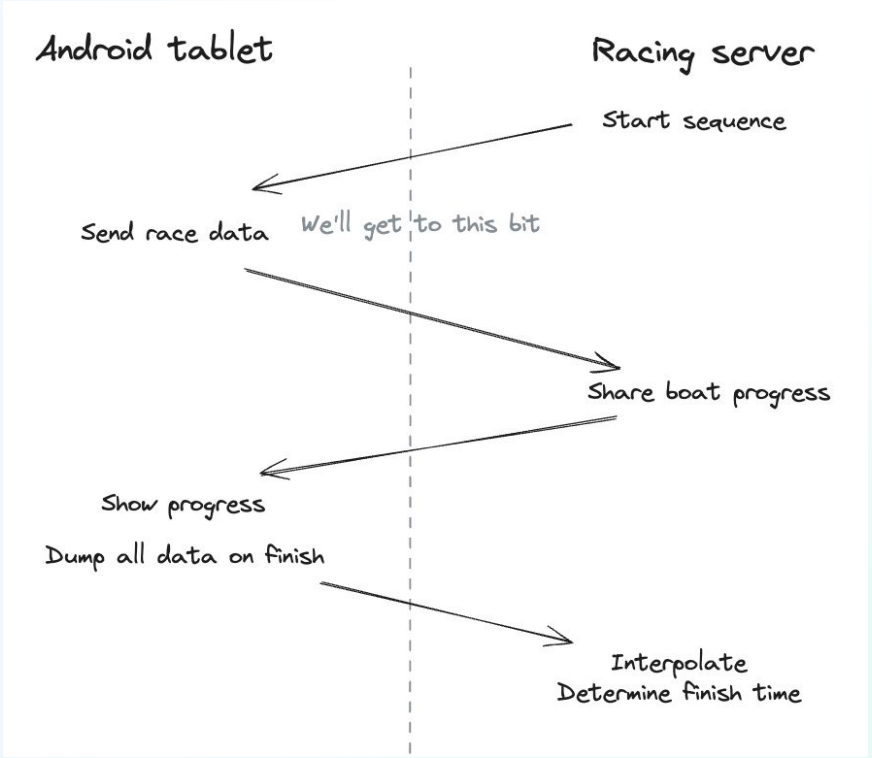
- Dumping data over a socket

```
'race/:raceId/boat/:boatId/participant/:participantId/progress'
```

- Combine data server side
  - Track progress of all devices
  - Calculate boat progress on every update
  - Buffer based on latency

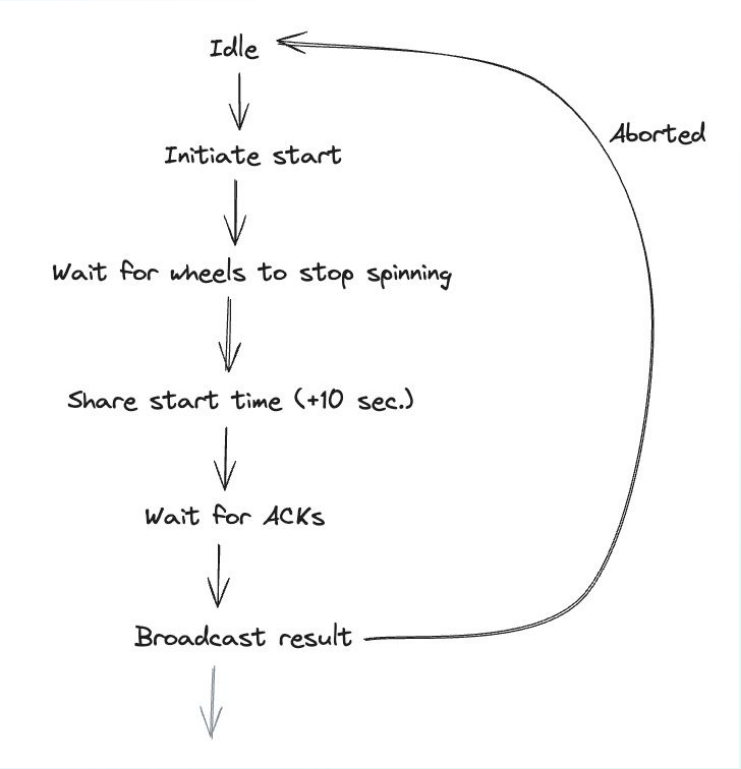
- Broadcast

```
`race/${raceId}/boat-progress`
```



The racing itself - start sequence

- All ergometers should be connected
- Every tablet should be ready
- No wheels should be spinning
- False starts
- Manual aborts at any time



The racing itself - start sequence

- Event based streams through Rx

Example: timer that will log every even second

```
const subscription = interval(1000)
   .pipe(
      filter((value) => value % 2 === 0),
      map((value) => `Elapsed time: ${value} seconds`),
   )
   .subscribe(console.log);
```

The racing itself - start sequence

```
from(this.raceProvider.get(raceId)).pipe(
   mergeMap((race) => this.wheelSpinnerWaiter.waitForWheelsToStop(race)),
   mergeMap((race) => this.startTimeChooser.choose(race)),
   mergeMap((race) => this.startIntentInteractor.execute(race)),
   mergeMap((race) => this.raceEntitiesProvider.get(race.id)),
   mergeMap((entities) => this.waitForAcks(entities.race, entities.boats, entities.participants)),
   map(([ackResult, race]) =>
       ackResult.success
           ? this.broadcastSuccessResult(race)
           : this.broadcastAckErrorResult(ackResult, race),
   ),
);
```

The racing itself - start sequence

```
merge(
    this.abortStartResultsCollector.collect(raceId),
    this.startFlowManager.manage(raceId),
)
    .pipe(first())
    .subscribe(console.log);
```

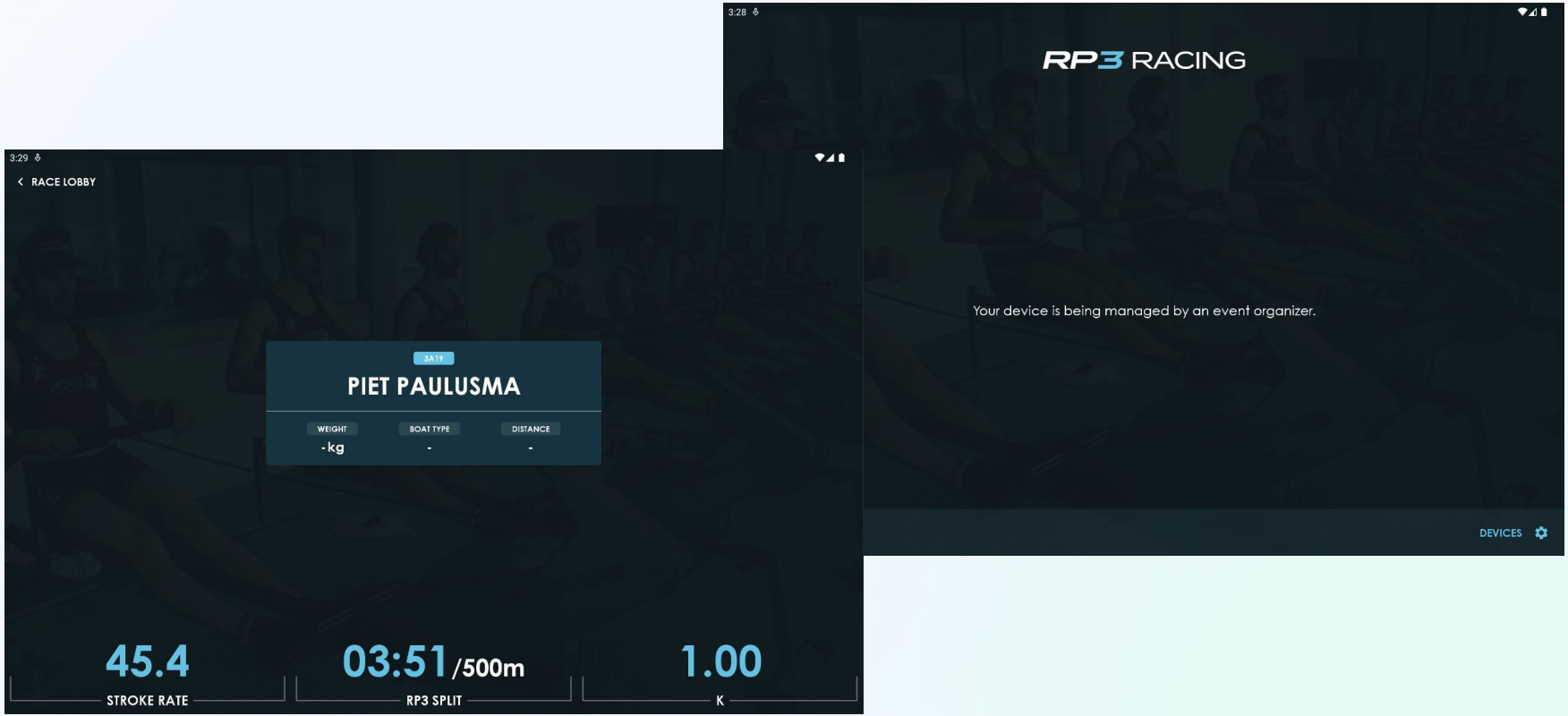
Back to the event



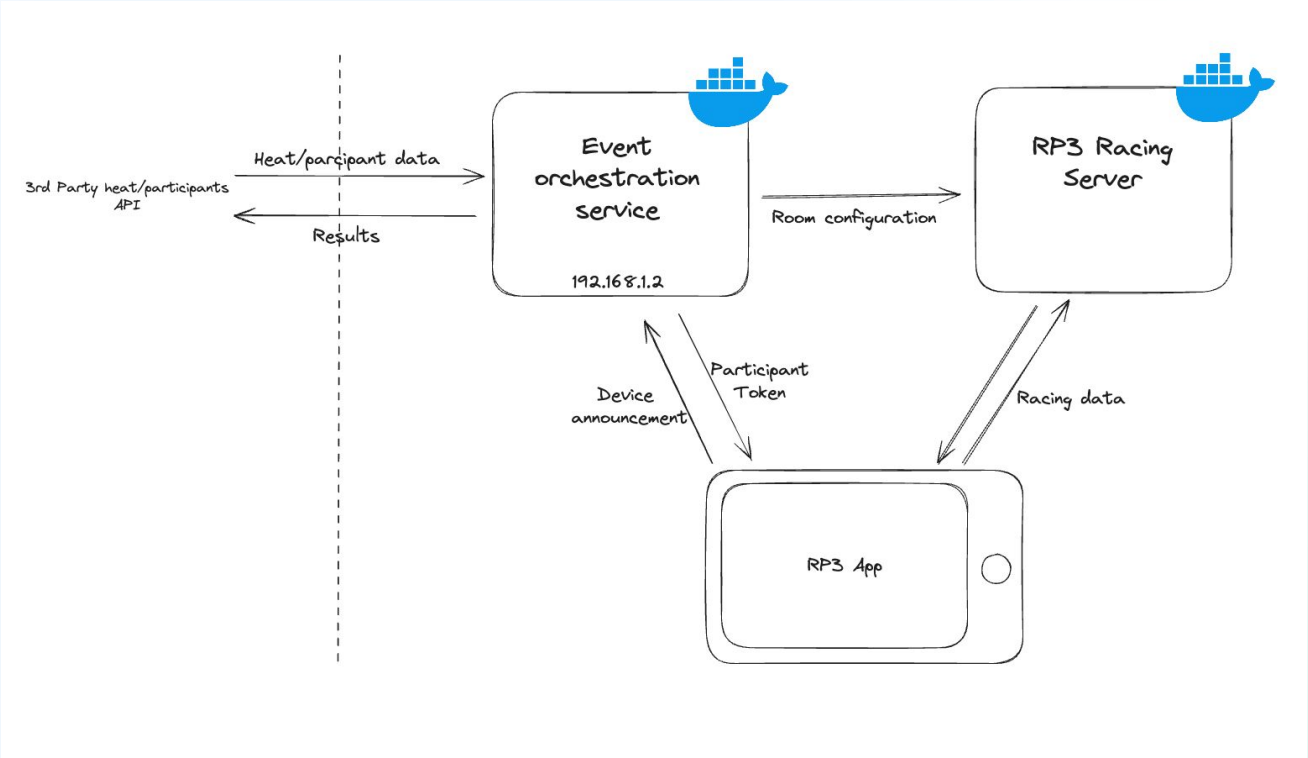
The orchestration

- What if you need to control 60+ tablets?
- Require 100% uptime without external dependencies
- Guaranteed capacity
- Run the event in the middle of nowhere (or the USA)

The orchestration

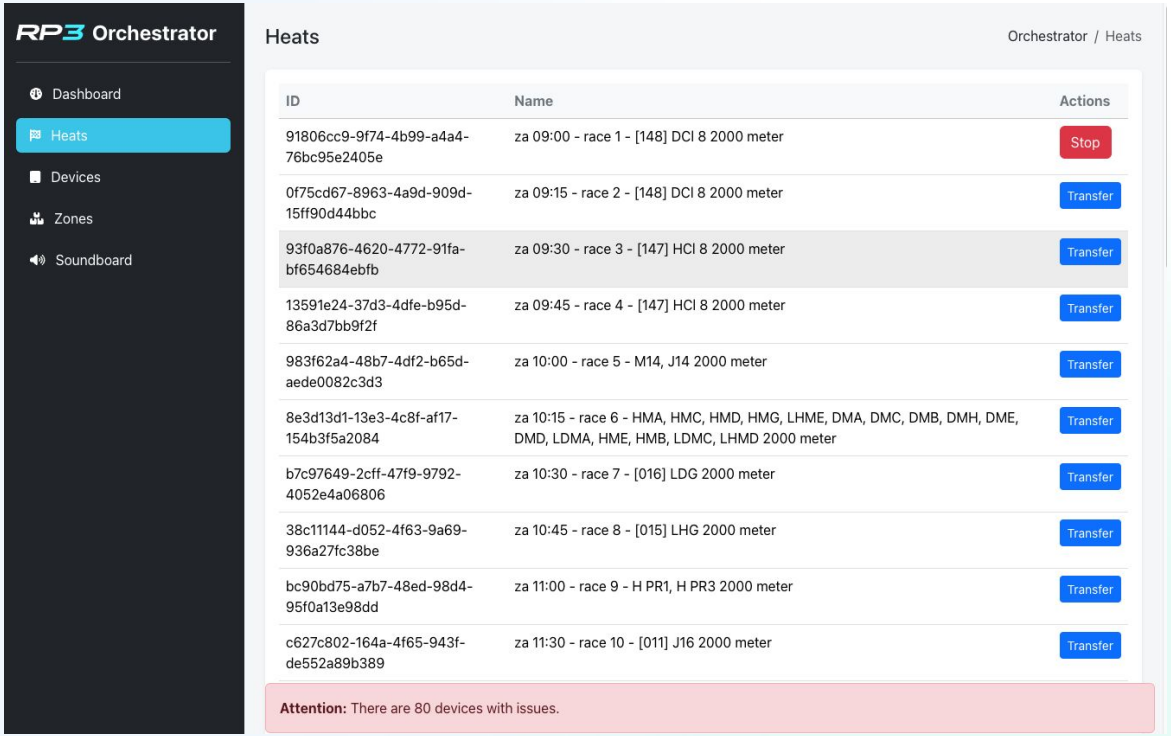


The orchestration



The orchestration

- Headless server
- Node.js back-end
- React UI
- Docker container



The UI



The UI


RP3
0:12.60
RACE OVERVIEW
STROKE RATES
1 MCBOATFACE 0.1M 21.9
2 MCBOATFACE 0.2S 20.9
3 MCBOATFACE 0.7X 20.4
4 MCBOATFACE 0.9B 20.1
5 MCBOATFACE 0.GM 20.6
6 MCBOATFACE 0.LU 21.3
7 MCBOATFACE 0.OS 20.9
8 MCBOATFACE 0.PG 21.9
9 MCBOATFACE 0.RV 21.6
10 MCBOATFACE 0.V5 22.0
GO!
4
5
6
7
8
9
10

The UI


RP3
0:12.69
RACE OVERVIEW
STROKE RATES
1 MCBOATFACE 0.1M 21.9
2 MCBOATFACE 0.2S 20.9
3 MCBOATFACE 0.7X 20.4
4 MCBOATFACE 0.9B 20.1
5 MCBOATFACE 0.GM 20.6
6 MCBOATFACE 0.LU 21.3
7 MCBOATFACE 0.OS 20.9
8 MCBOATFACE 0.PG 21.9
9 MCBOATFACE 0.RV 21.5
10 MCBOATFACE 0.VS 22.0
GO!
4 5 6 7 8 9 10

The UI

- Socket.IO for data streams

```
import {io} from 'socket.io-client';
import {fromEvent, Observable} from 'rxjs';

const socket = io('/');
fromEvent(socket, `race/${raceId}/boat-progress`);
```

The UI

- React Konva for canvas drawing

```
import {Layer, Rect, Stage} from 'react-konva';

<Stage width={canvasSizes.canvasWidthPx} height={canvasSizes.canvasHeightPx}>
   <Layer>
       <Rect ... />
       <Lanes ... />
       <BuoyLanes ... />
       <LaneNumbers ... />
       <BigBuoys ... />
       <WaterContent ... />
       <AnimatingBoats ... />
   </Layer>
   <Overlays ... />
</Stage>
```

The UI

- Separate boats from camera

```
<BuoyLanes
   viewportOffsetPx={cameraOffsetPx}
   lanes={lanes}
/>
```

```
interface CameraOffsetCalculator {
   calculate(
       boatsWithCombinations: BoatWithCombinations[], //The boats that should be rendered
       realtimeBoatProgresses: RealtimeBoatProgress[], //The progress of each boat
       canvasSizes: CanvasSizes,
   ): number;
}
```

The UI

- Rx animation frame scheduler
- Debounce any UI updates till the next frame

```
const animationFrameScheduler = new AnimationFrameScheduler(AnimationFrameAction);

export function debounceThrottleAnimationFrame<T>(obs: Observable<T>, minDuration: number = 0):
Observable<T> {
   return merge(obs.pipe(first()), obs.pipe(auditTime(minDuration, animationFrameScheduler)));
}
```

MoveLab

Thank you!



Thank you for joining our Meetup

El Niño Meetups

Get involved! — Scan the QR code on the left to give us some feedback and/or join our Meetup community.

EL NIÑO