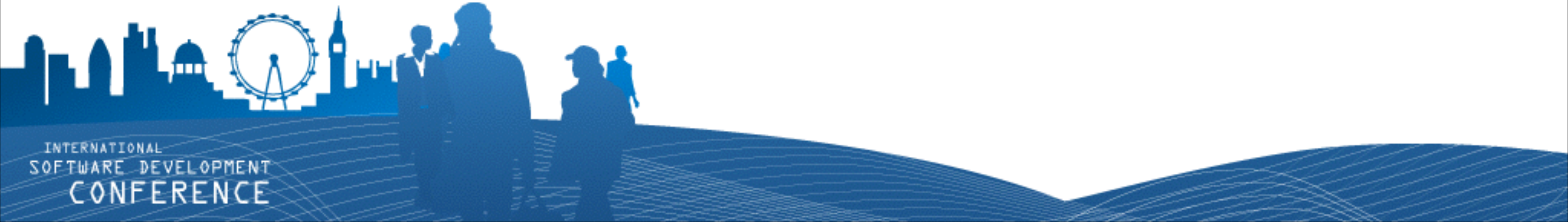
QCon

Please evaluate my talk via the mobile app!



Sensors aren’t enough

Mo Ramezanpoor
Six to Start
@mohsenr

Agenda

A look at fitness tracking apps

- Tailored data processing
- Adapting the UI

Overcoming device limitations

- Reducing battery use
- Keeping the app alive

Handling sensor data

- Keeping track of time
- Distance calculation

2
SIX TO START & NAOMI ALDERMAN

THE WALK
SIX TO START & NAOMI ALDERMAN

Zombies, Run!

The Walk

Reducing battery use

Reducing battery use

Don’t use CPU if you can

- Get data points in batches
- Don’t use standalone timers to update the UI
- Defer data processing as much as possible

Only ask for the accurate that you need

- Accurate location data will require multiple antennae
- Accurate motion data will require multiple sensors
- Accurate data points are computationally more expensive to produce

Keeping the app alive (iOS)

Keeping the app alive (iOS)

Play audio in the background

- If you already need to play audio in the background
- If you need to completely stop the device from sleeping

Track location

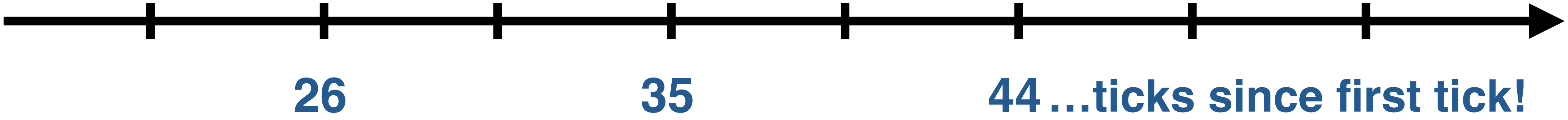
- Using significant location change updates
- Using explicit location tracking but with low accuracy

Use notifications

- If your app's behaviour depend on external sources
- f you have can assume network connectivity

Keeping track of time

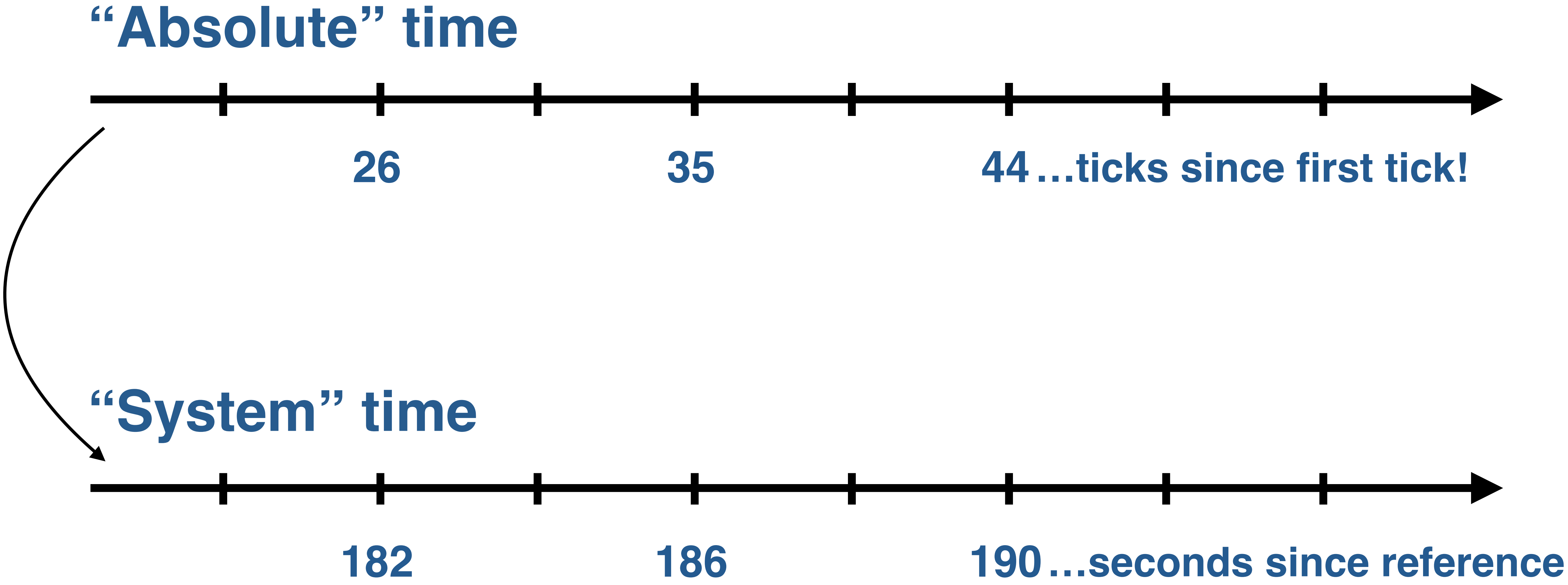
“Absolute” time



26

35

44…ticks since first tick!



“Absolute” time

26

35

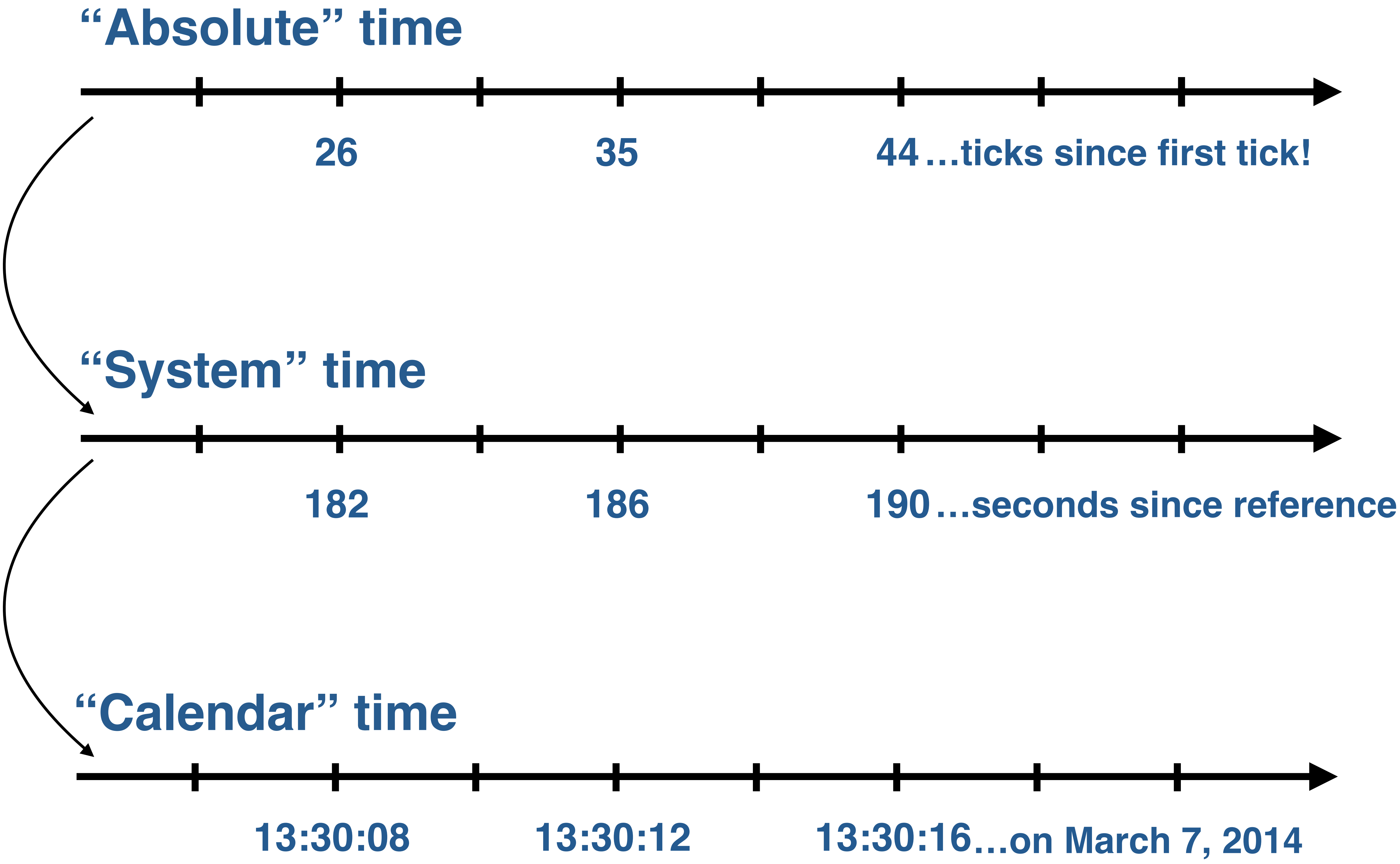
44…ticks since first tick!

“System” time

182

186

190…seconds since reference



“Absolute” time

26

35

44…ticks since first tick!

“System” time

182

186

190…seconds since reference

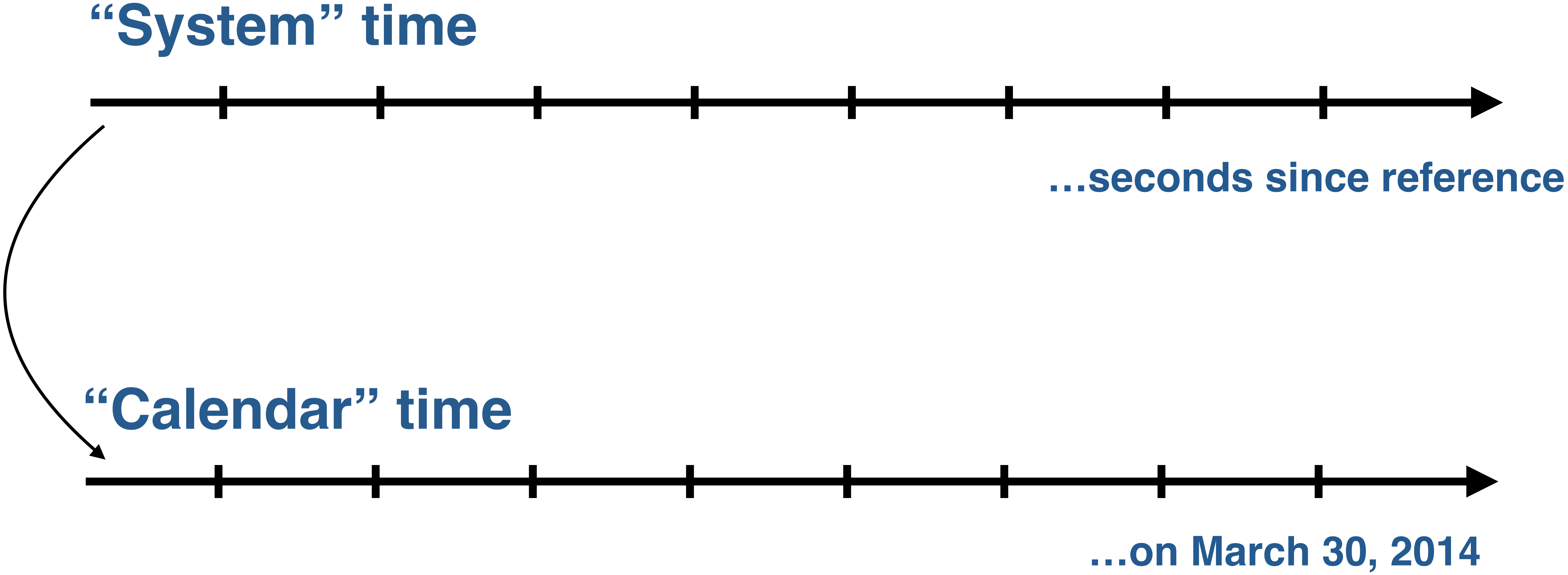
“Calendar” time

13:30:08

13:30:12

13:30:16…on March 7, 2014

Calendar time jumps



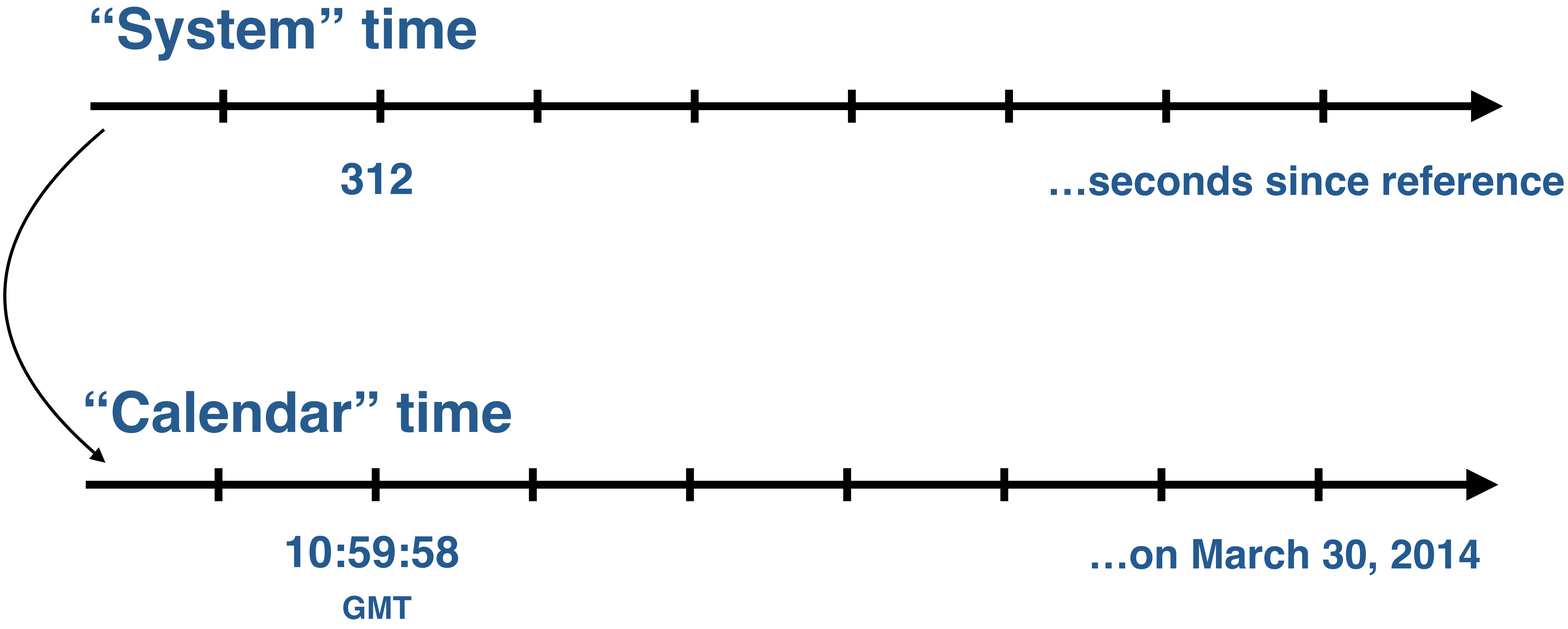
“System” time

…seconds since reference

“Calendar” time

…on March 30, 2014

Calendar time jumps



“System” time

312

…seconds since reference

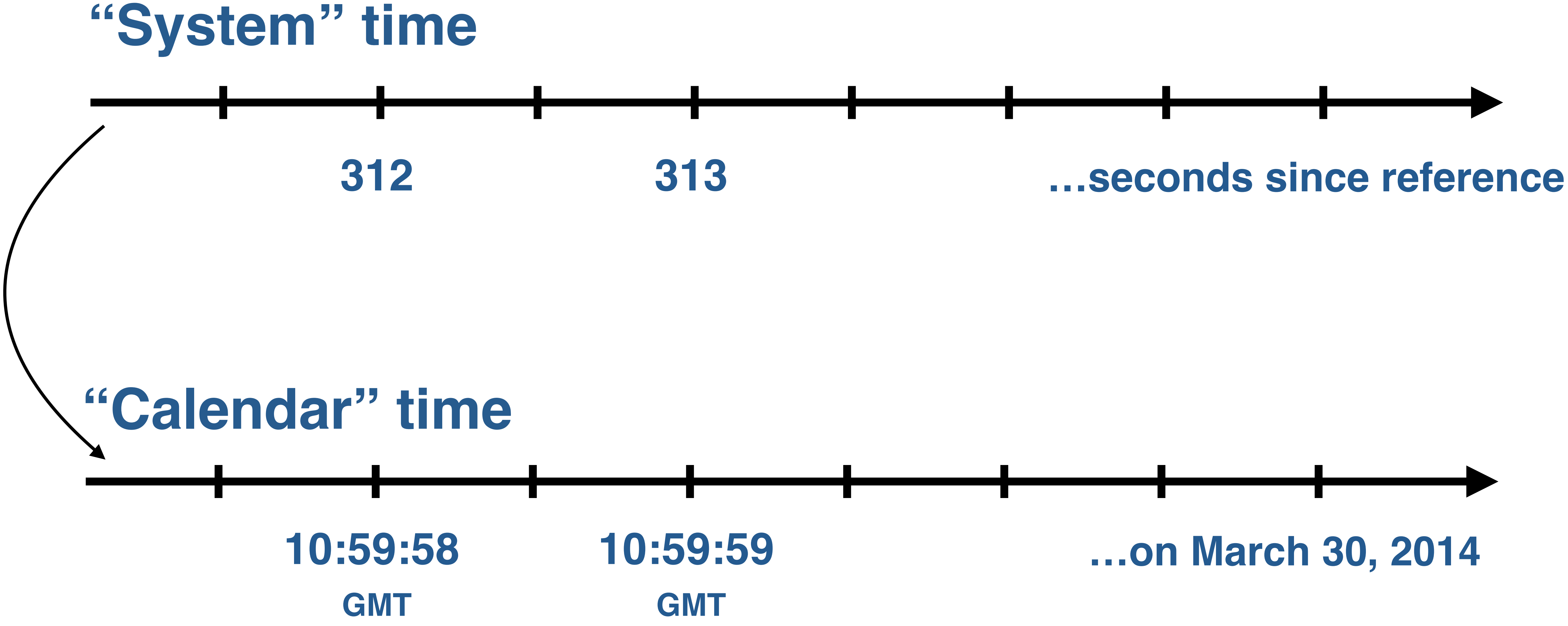
“Calendar” time

10:59:58

GMT

…on March 30, 2014

Calendar time jumps



“System” time

312

313

…seconds since reference

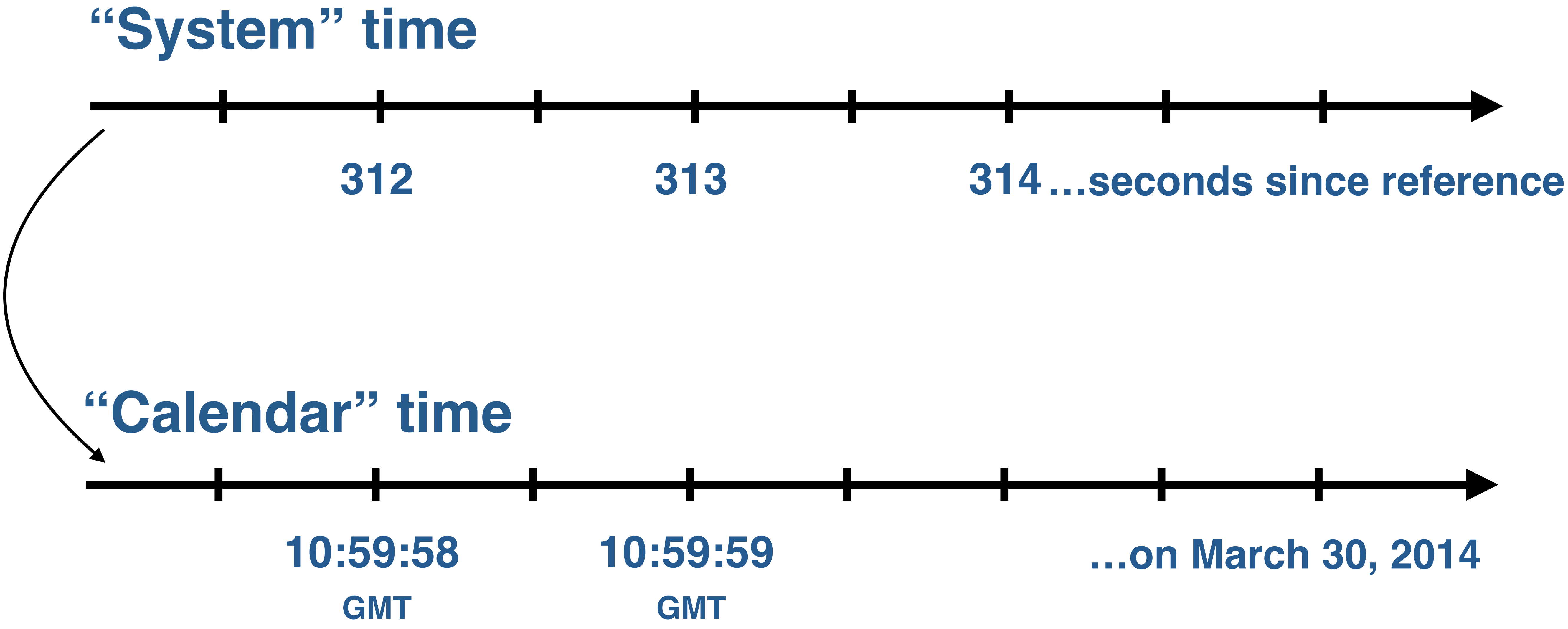
“Calendar” time

10:59:58
GMT

10:59:59
GMT

…on March 30, 2014

Calendar time jumps



“System” time

312

313

314…seconds since reference

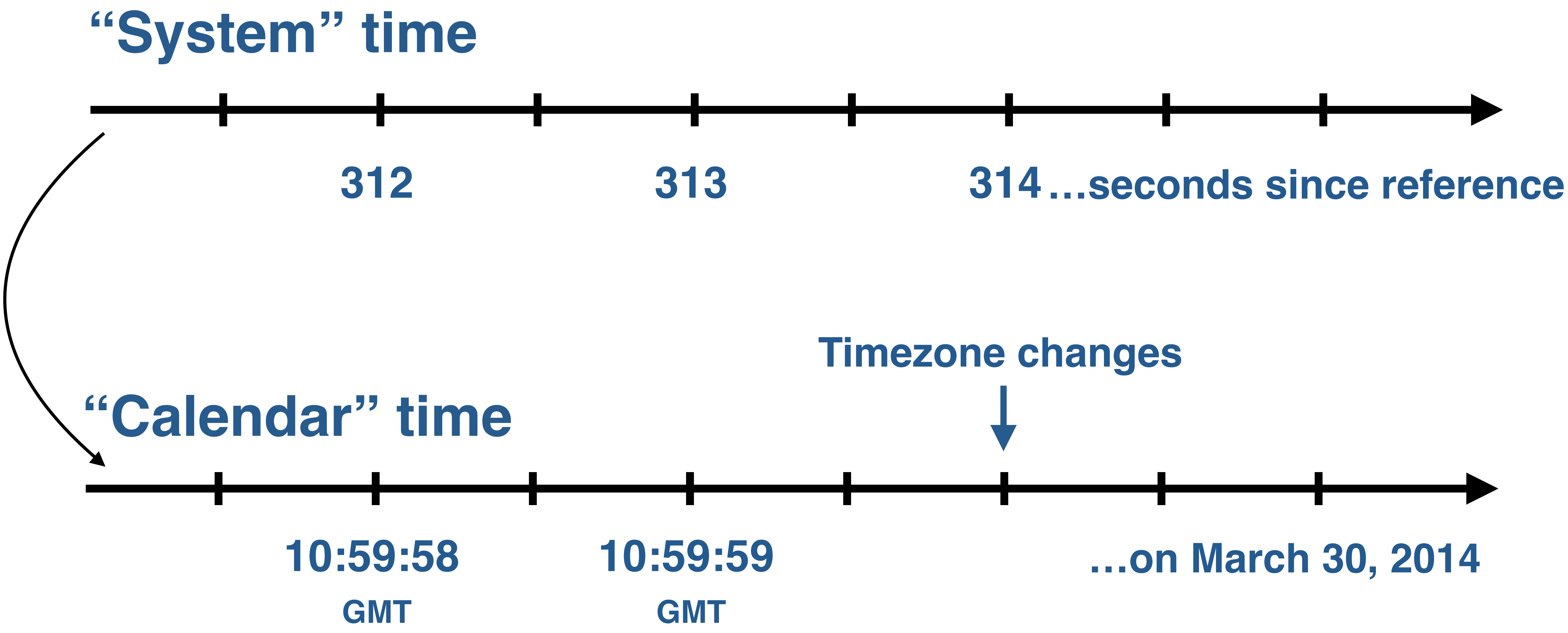
“Calendar” time

10:59:58 GMT

10:59:59 GMT

…on March 30, 2014

Calendar time jumps



“System” time

312

313

314…seconds since reference

Timezone changes

“Calendar” time

10:59:58

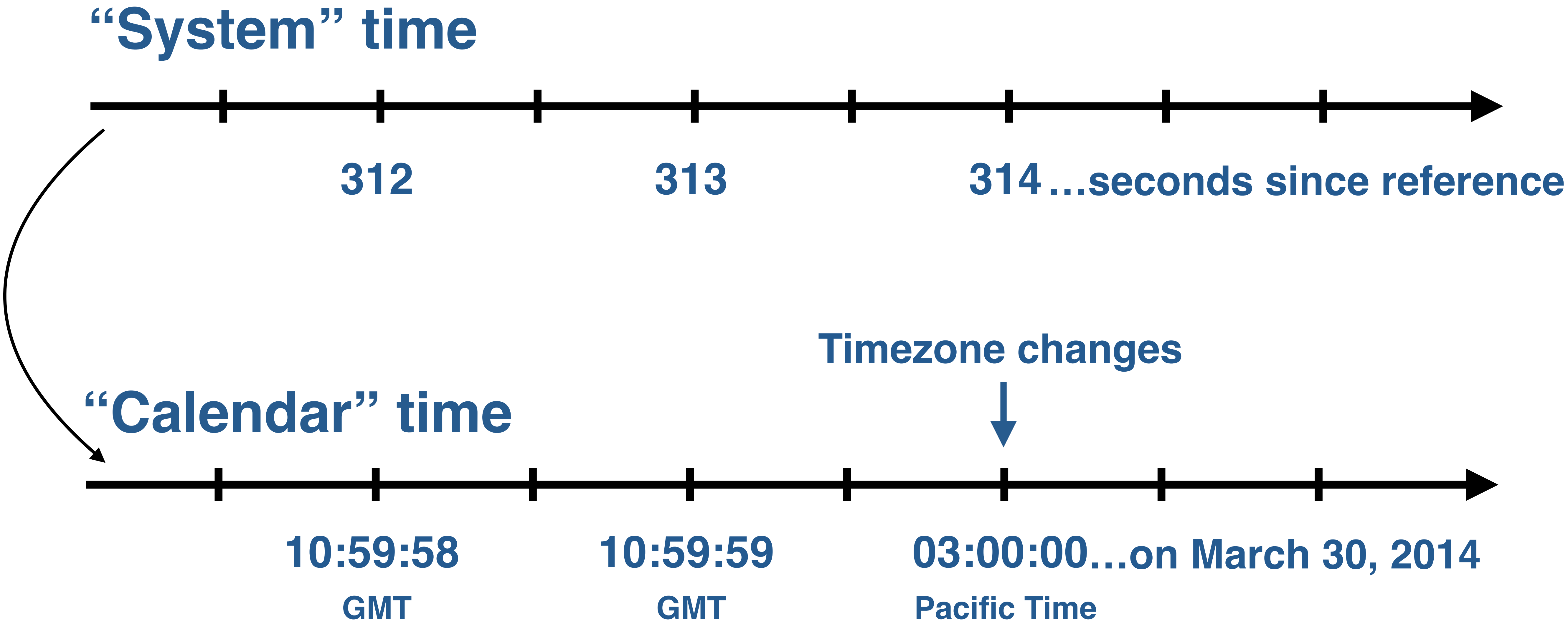
GMT

10:59:59

GMT

…on March 30, 2014

Calendar time jumps



“System” time

312

313

314…seconds since reference

Timezone changes

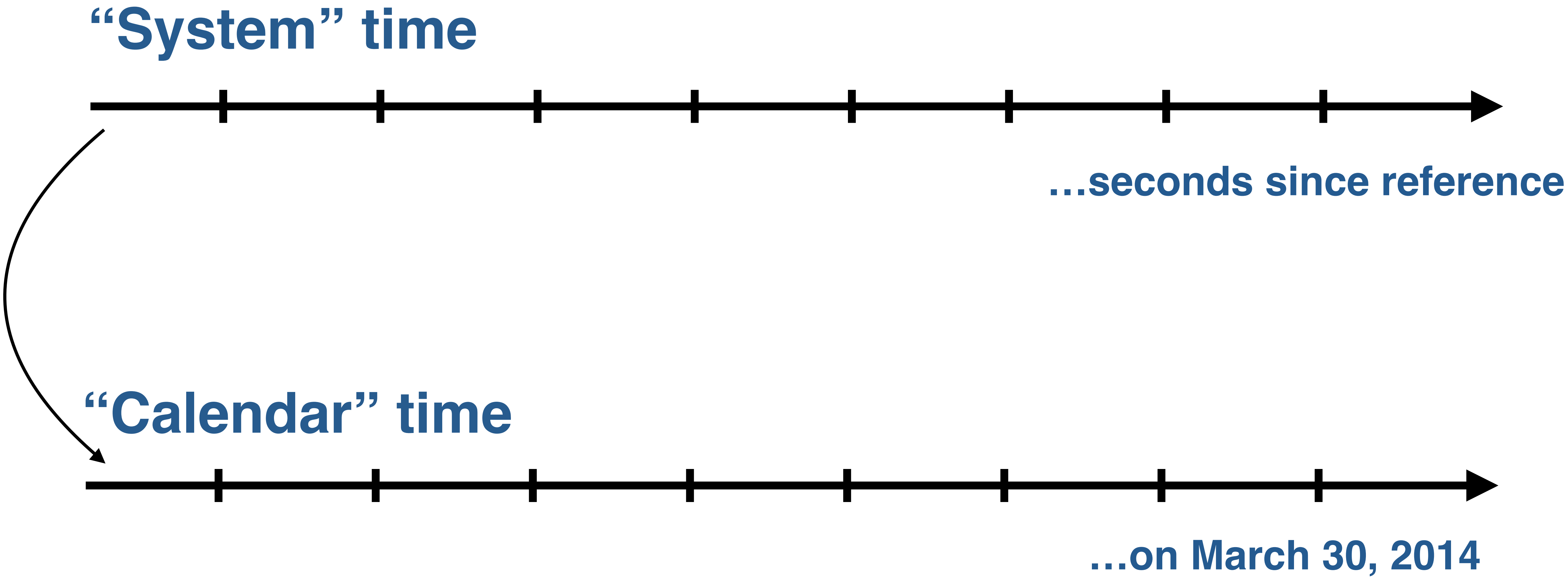
“Calendar” time

10:59:58
GMT

10:59:59
GMT

03:00:00…on March 30, 2014
Pacific Time

Calendar time jumps



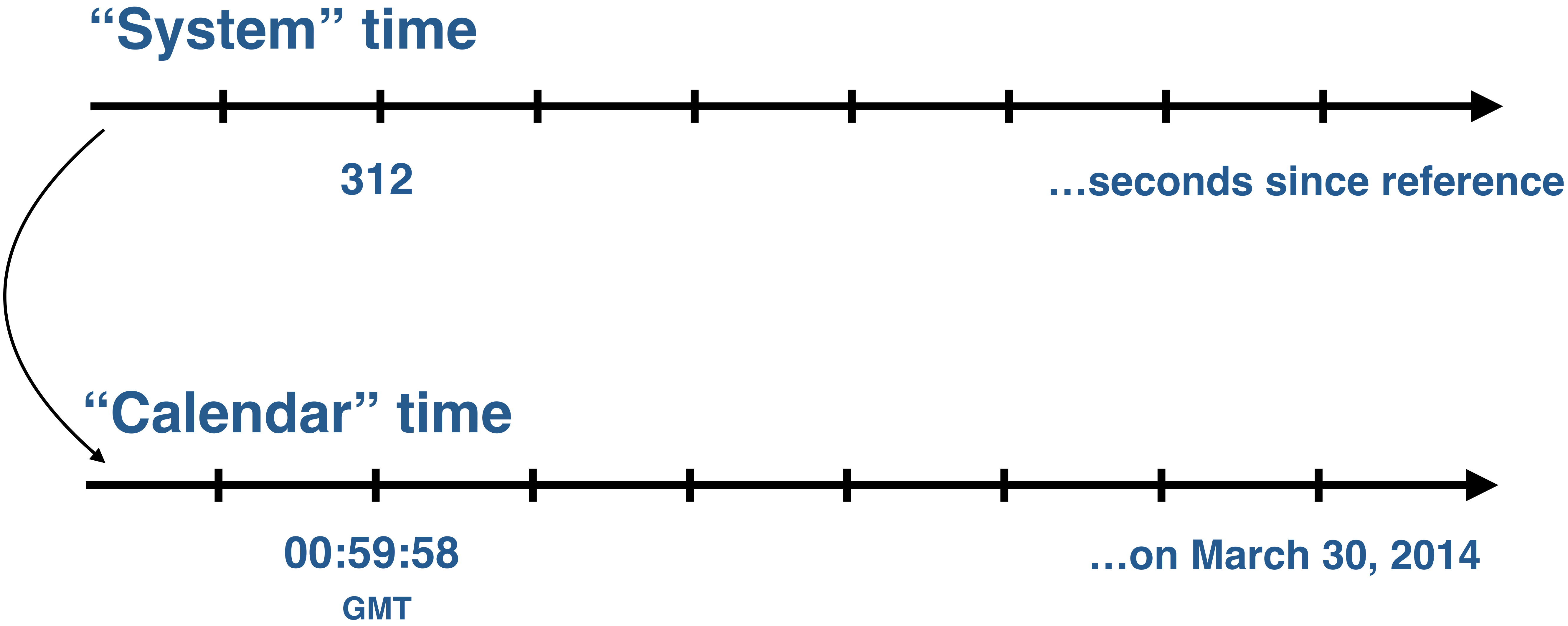
“System” time

…seconds since reference

“Calendar” time

…on March 30, 2014

Calendar time jumps



“System” time

312

…seconds since reference

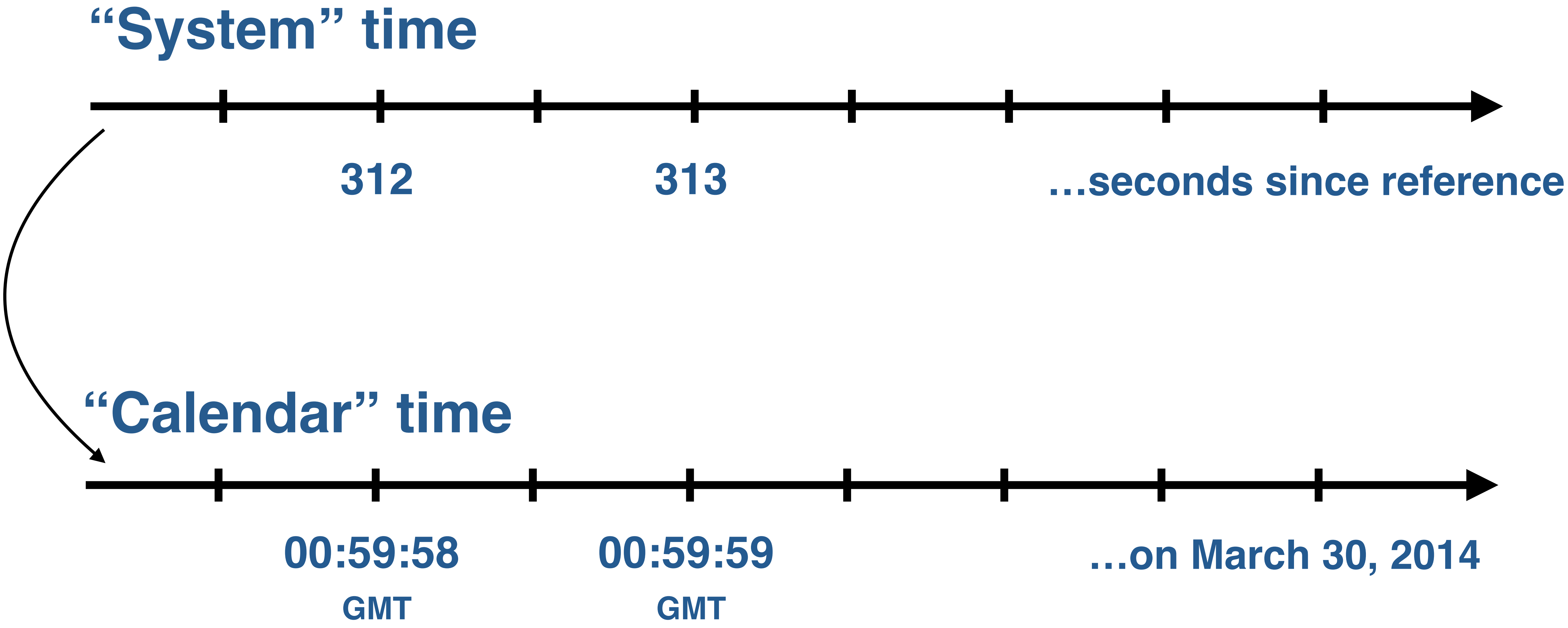
“Calendar” time

00:59:58

GMT

…on March 30, 2014

Calendar time jumps



“System” time

312

313

…seconds since reference

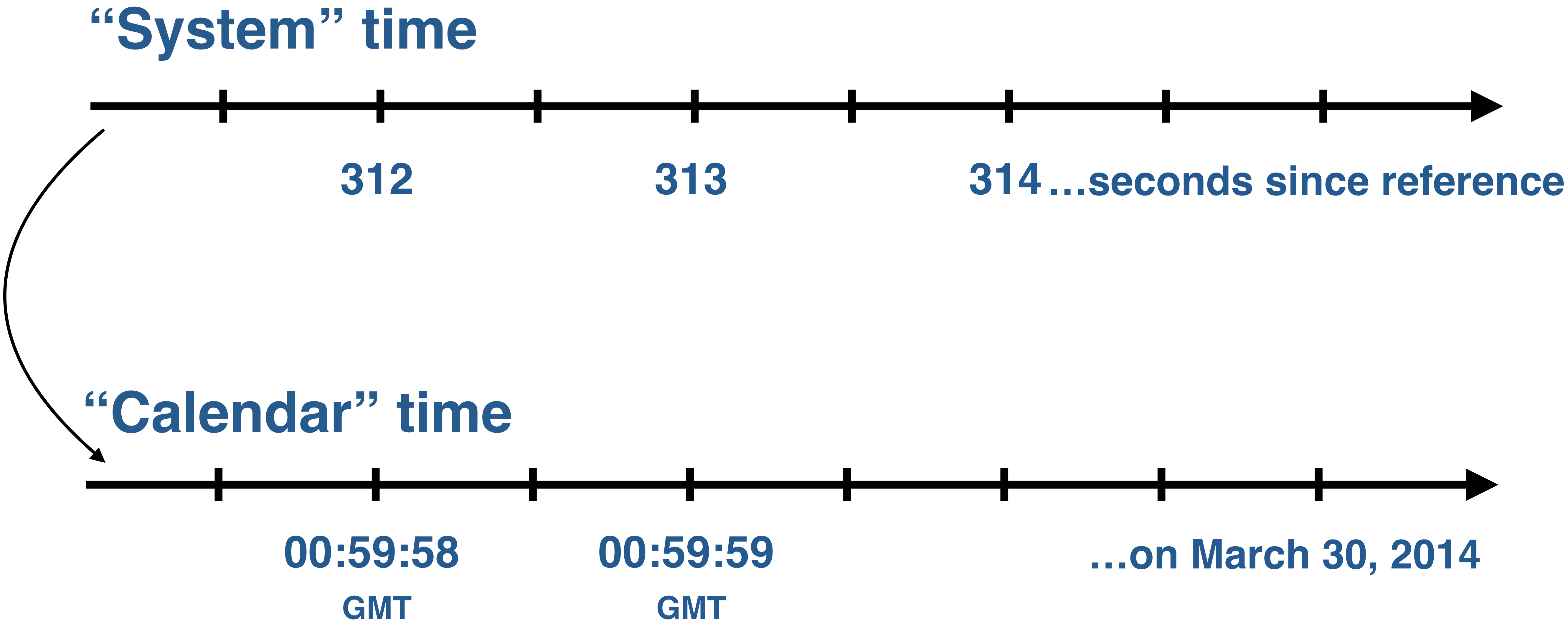
“Calendar” time

00:59:58
GMT

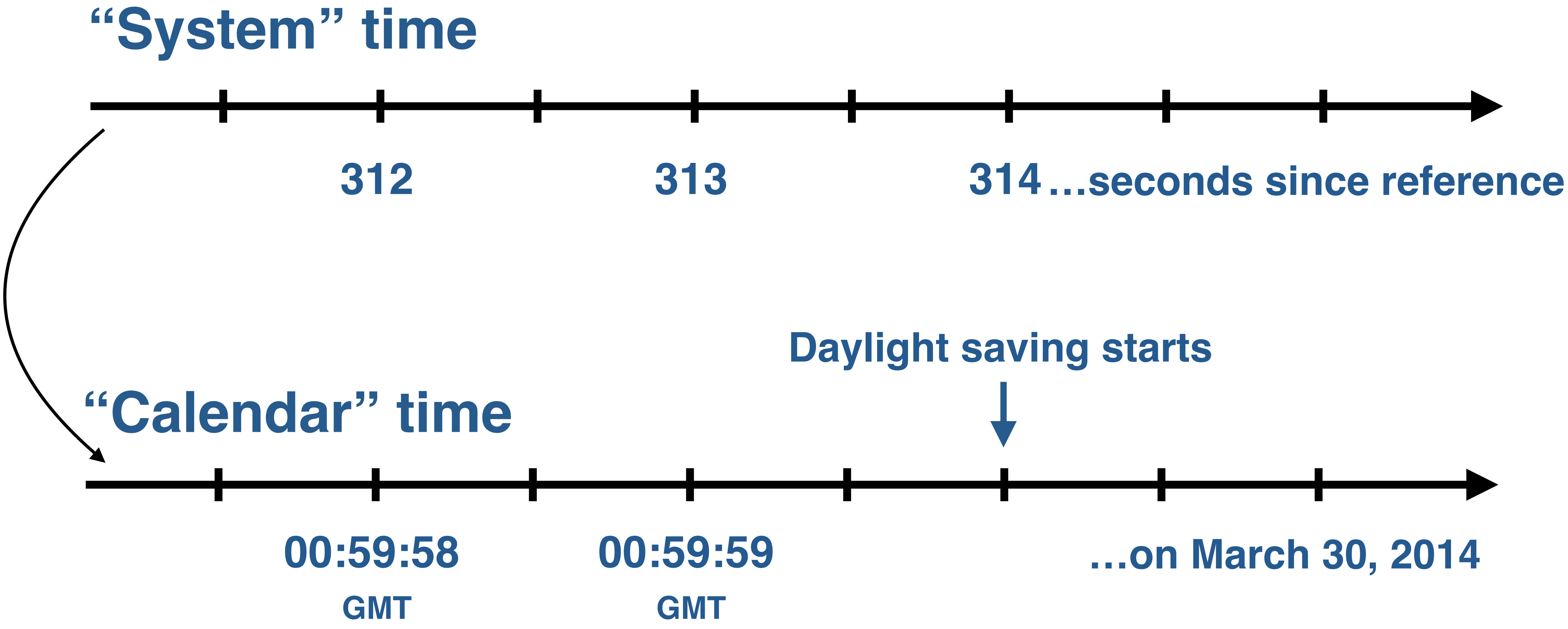
00:59:59
GMT

…on March 30, 2014

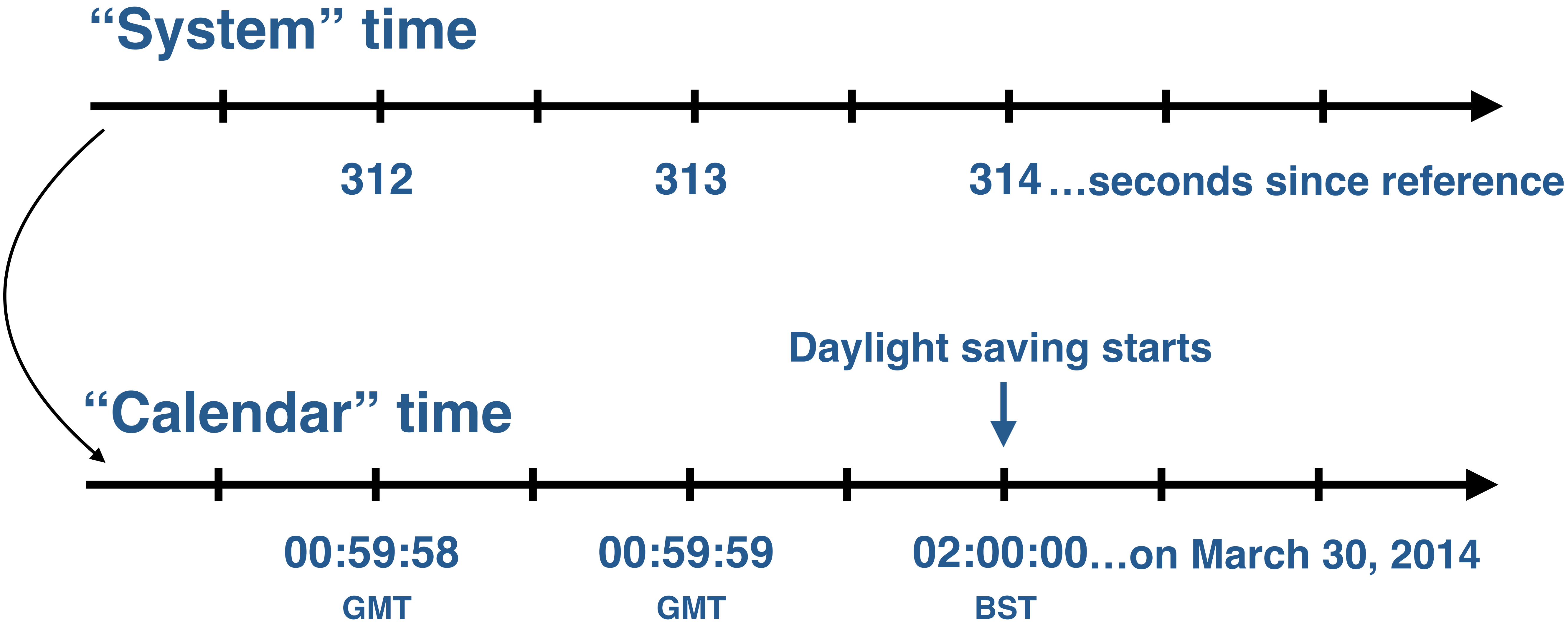
Calendar time jumps



Calendar time jumps



Calendar time jumps



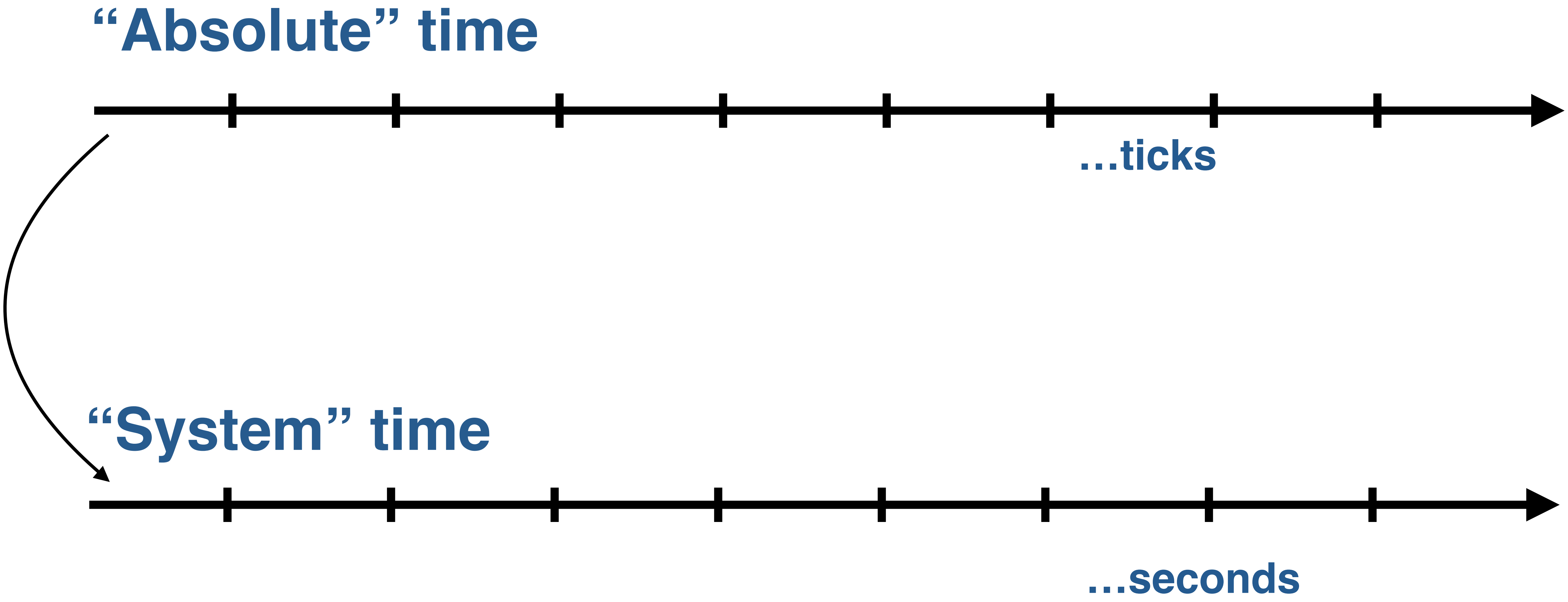
Calendar time jumps

Managing calendar date and time is hard

… but is not our problem

… as long as we don’t use them!

System time jumps



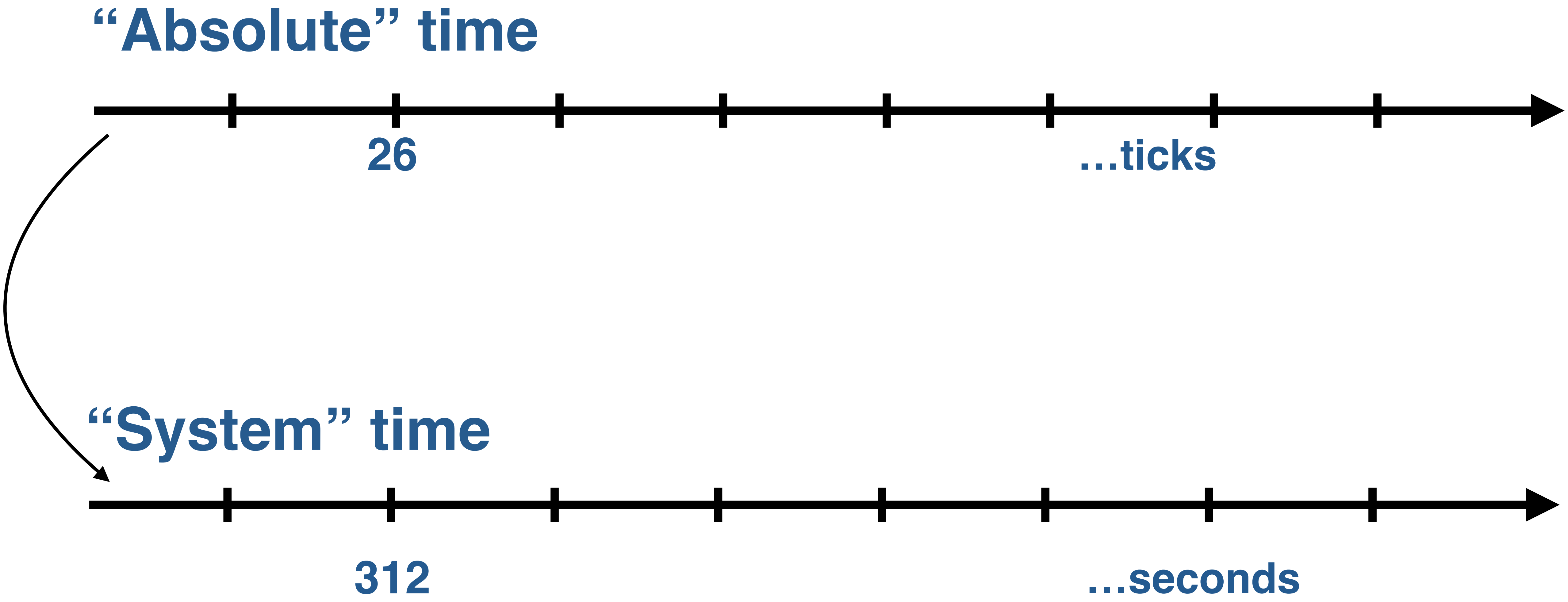
“Absolute” time

…ticks

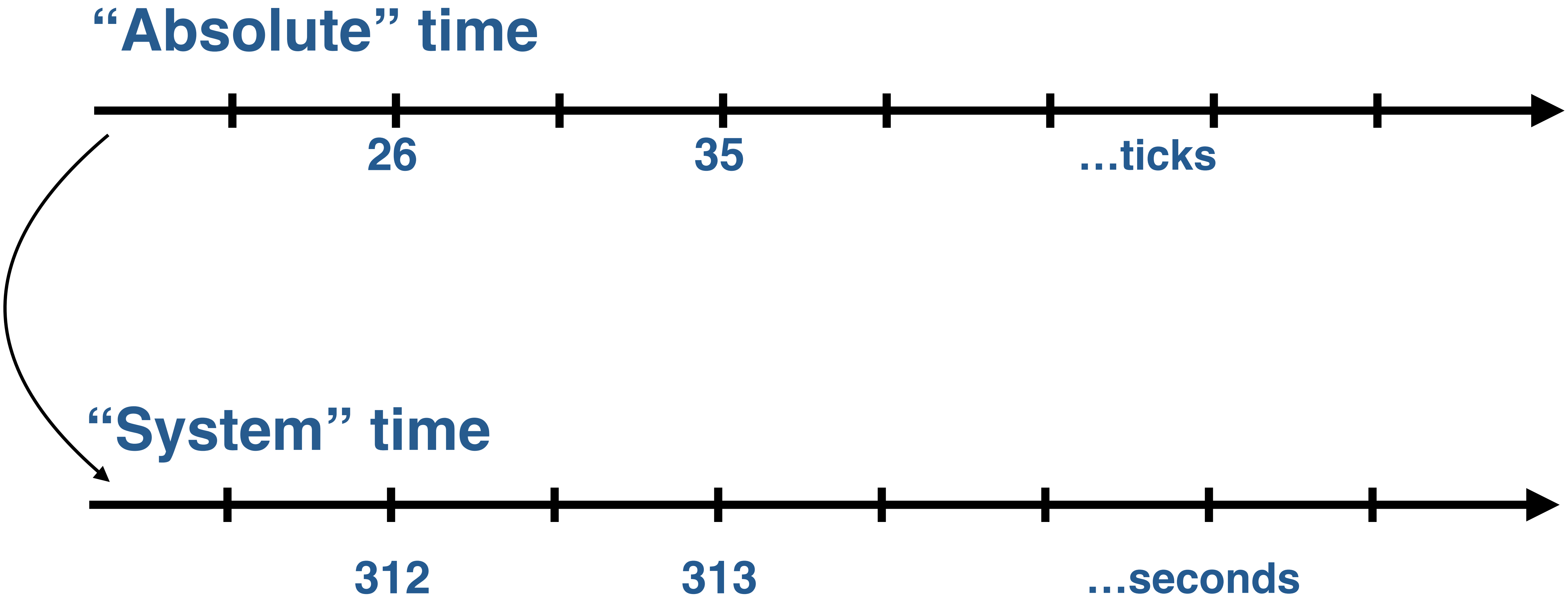
“System” time

…seconds

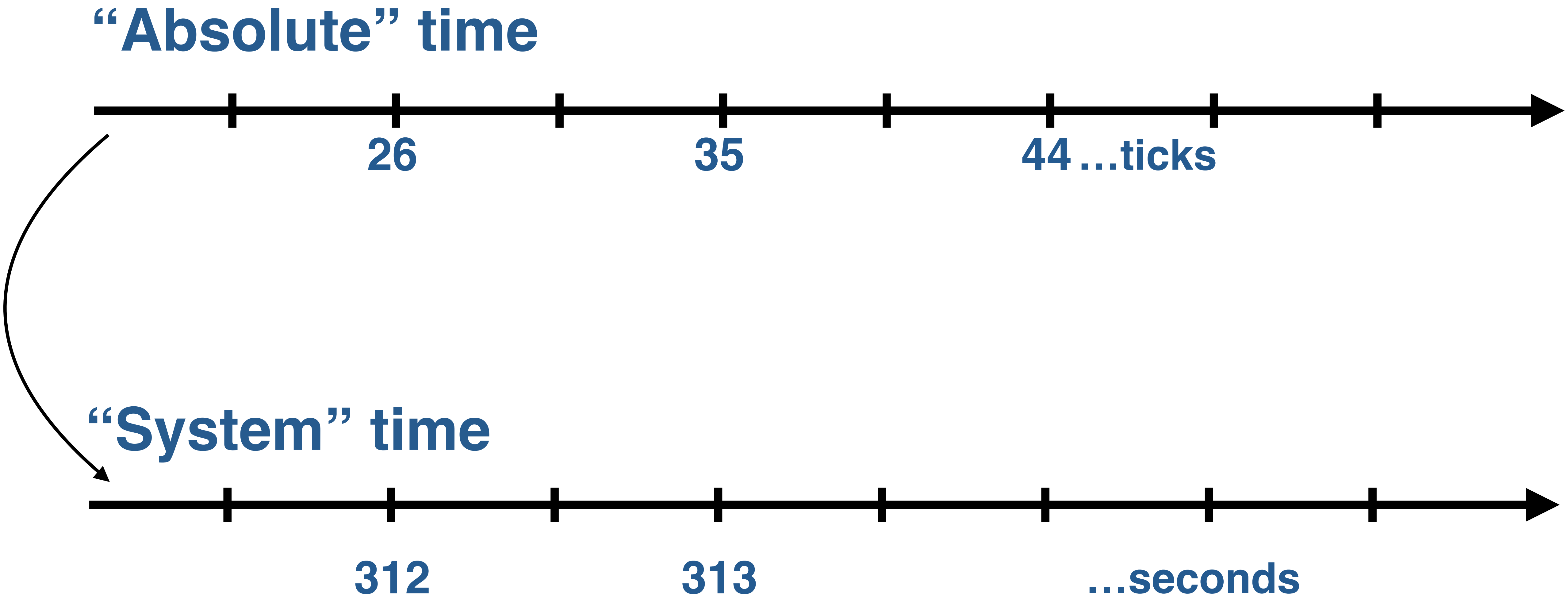
System time jumps



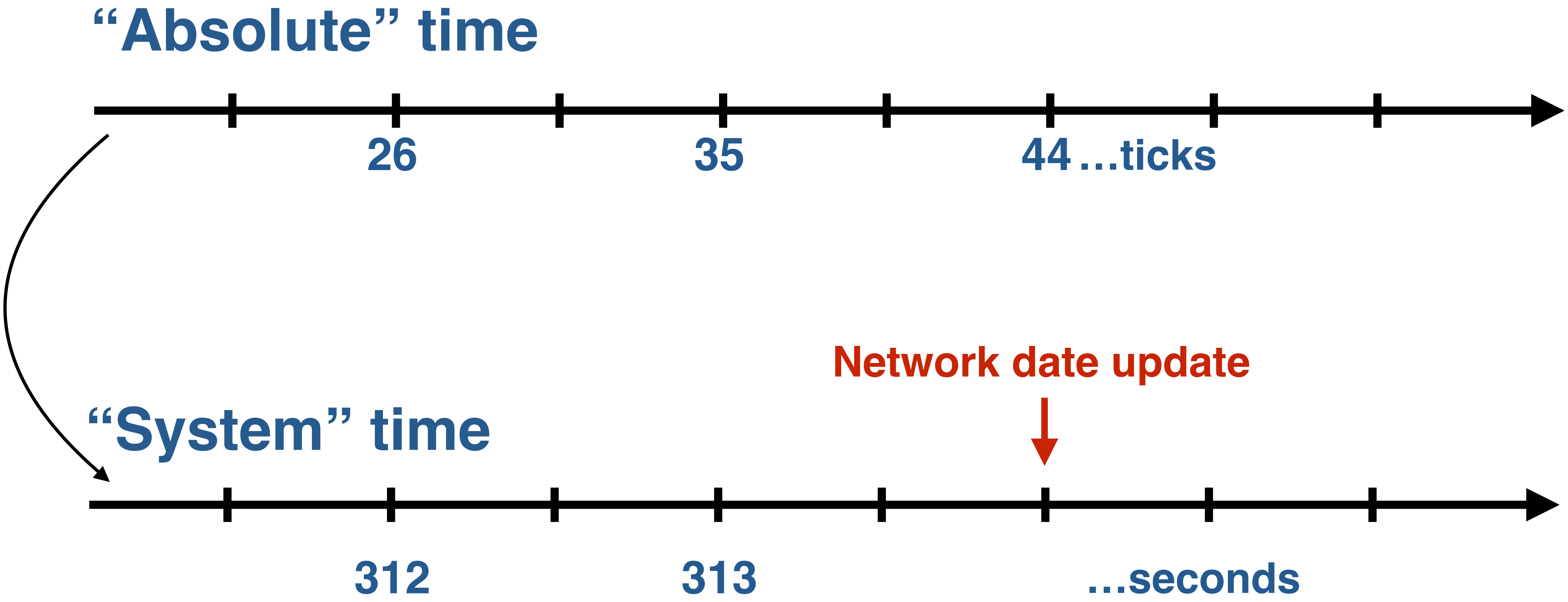
System time jumps



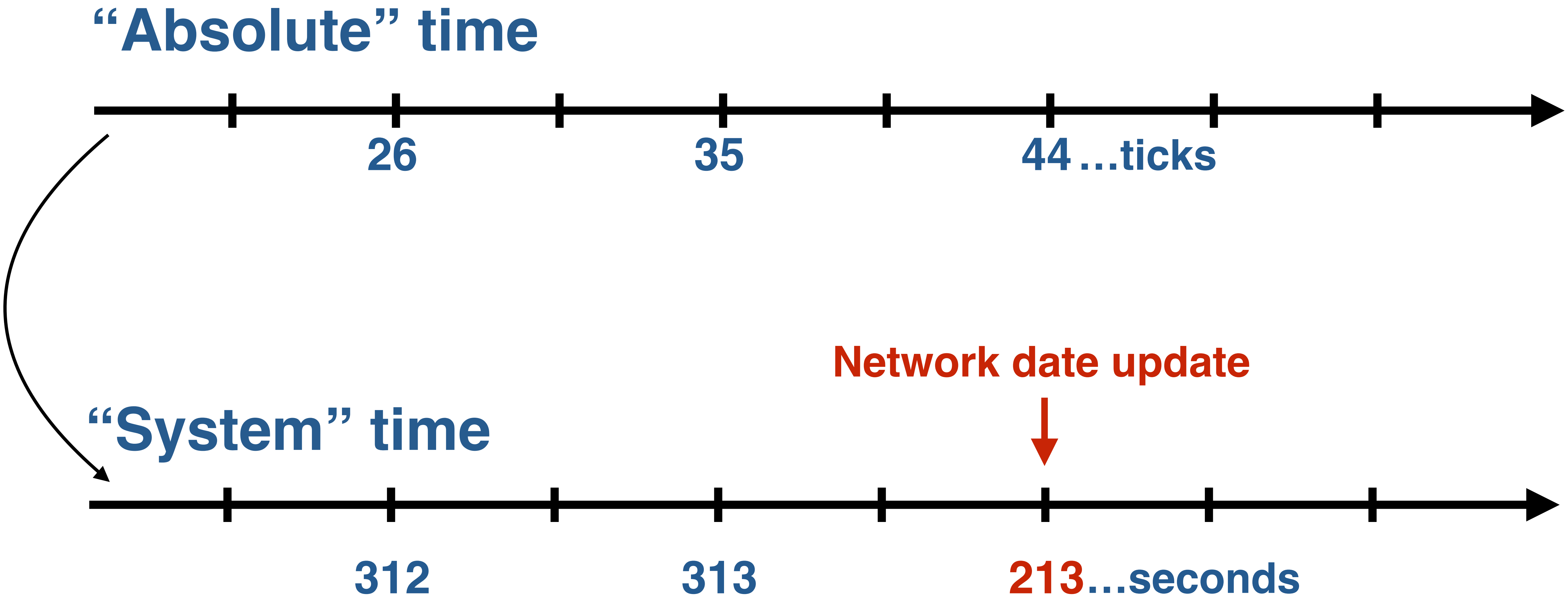
System time jumps



System time jumps



System time jumps



System time jumps

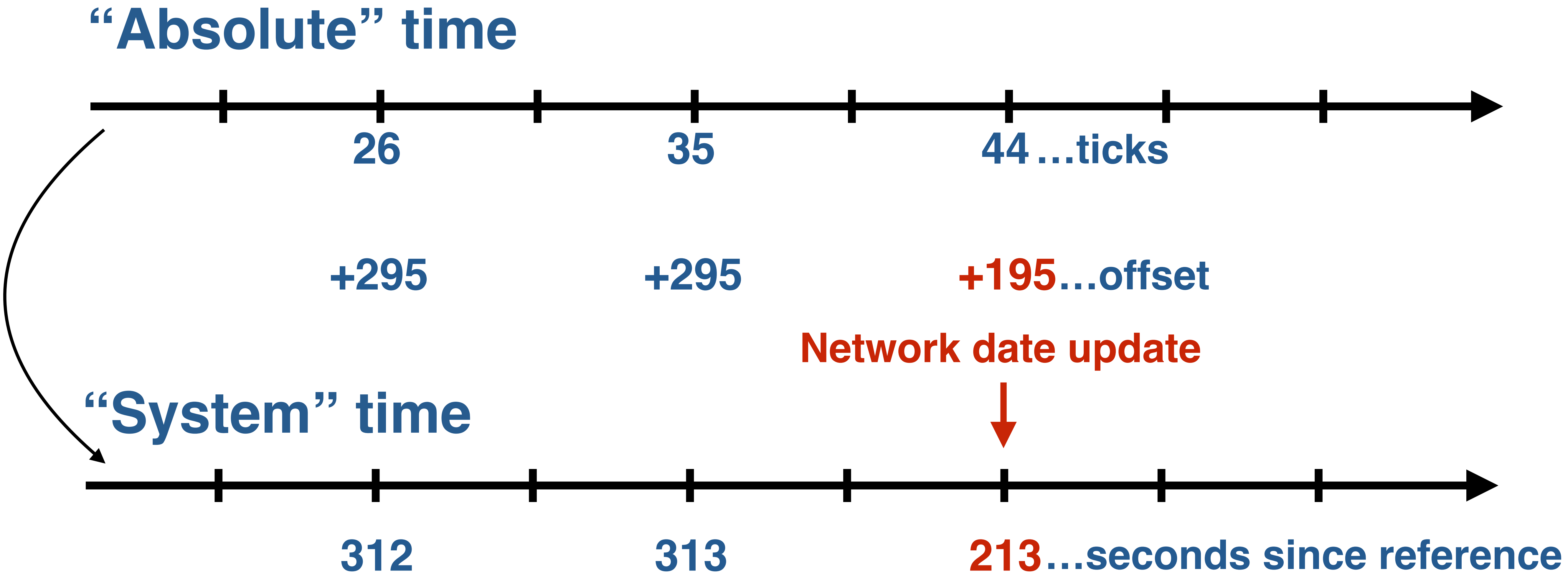
Solution:

1. Do nothing

2. “Anchor” the time

3. Tell the user

“Anchored” Time



“Absolute” time

26

35

44…ticks

+295

+295

+195…offset

Network date update

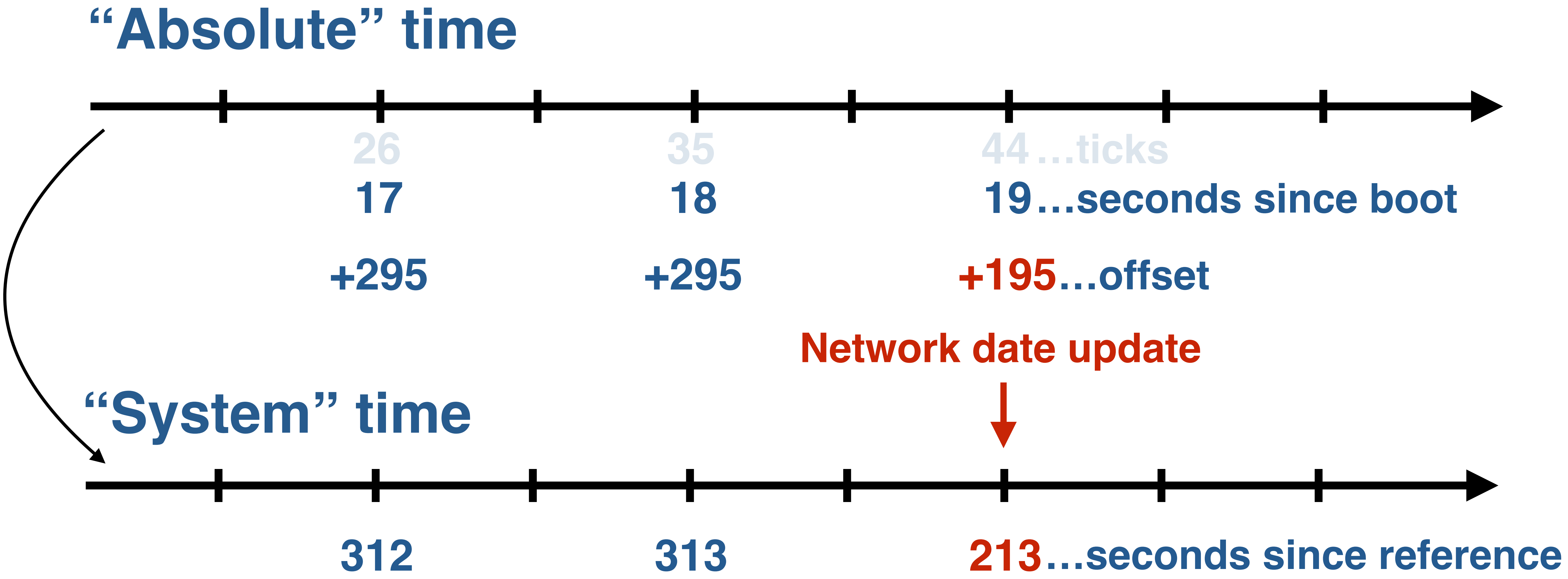
“System” time

312

313

213…seconds since reference

“Anchored” Time



“Anchored” Time

Prepare:

```
double initialOffset = SystemTime() - AbsoluteTime();
```

Start recording:

```
// Wait for data!
```

New data received:

```
double currentOffset = SystemTime() - AbsoluteTime();
double drift = currentOffset - initialOffset;
double anchoredTimestamp = timestamp - drift;
```

“Anchored” Time

On iOS:

- Use CACurrentMediaTime() as AbsoluteTime().
- Use [NSDate timeIntervalSinceReferenceDate] as SystemTime().

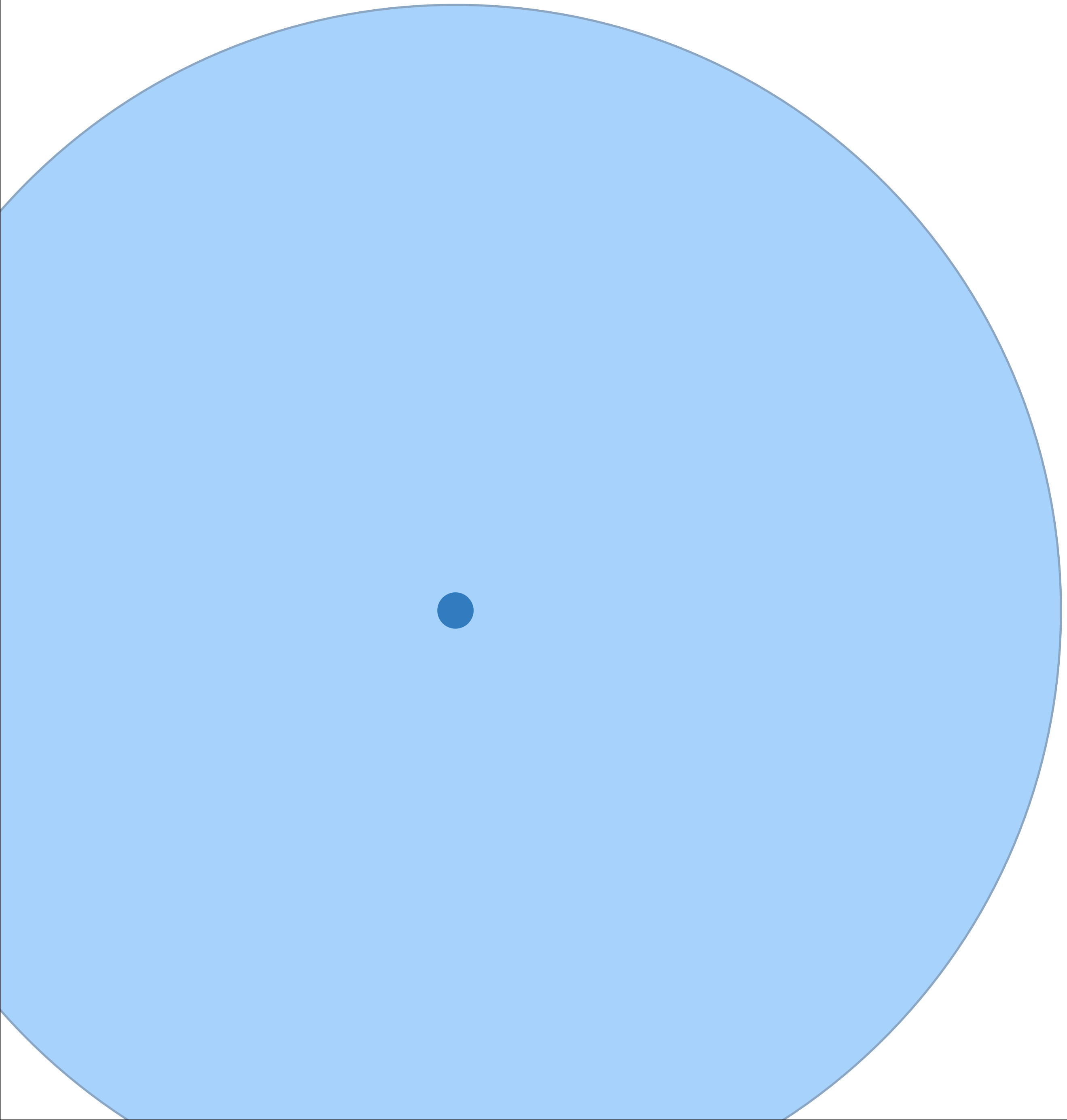
On Android:

- Use SystemClock.elapsedRealtimeNanos() as AbsoluteTime().
- Use System.currentTimeMillis() as SystemTime().
- Also see Location.getElapsedRealtimeNanos().

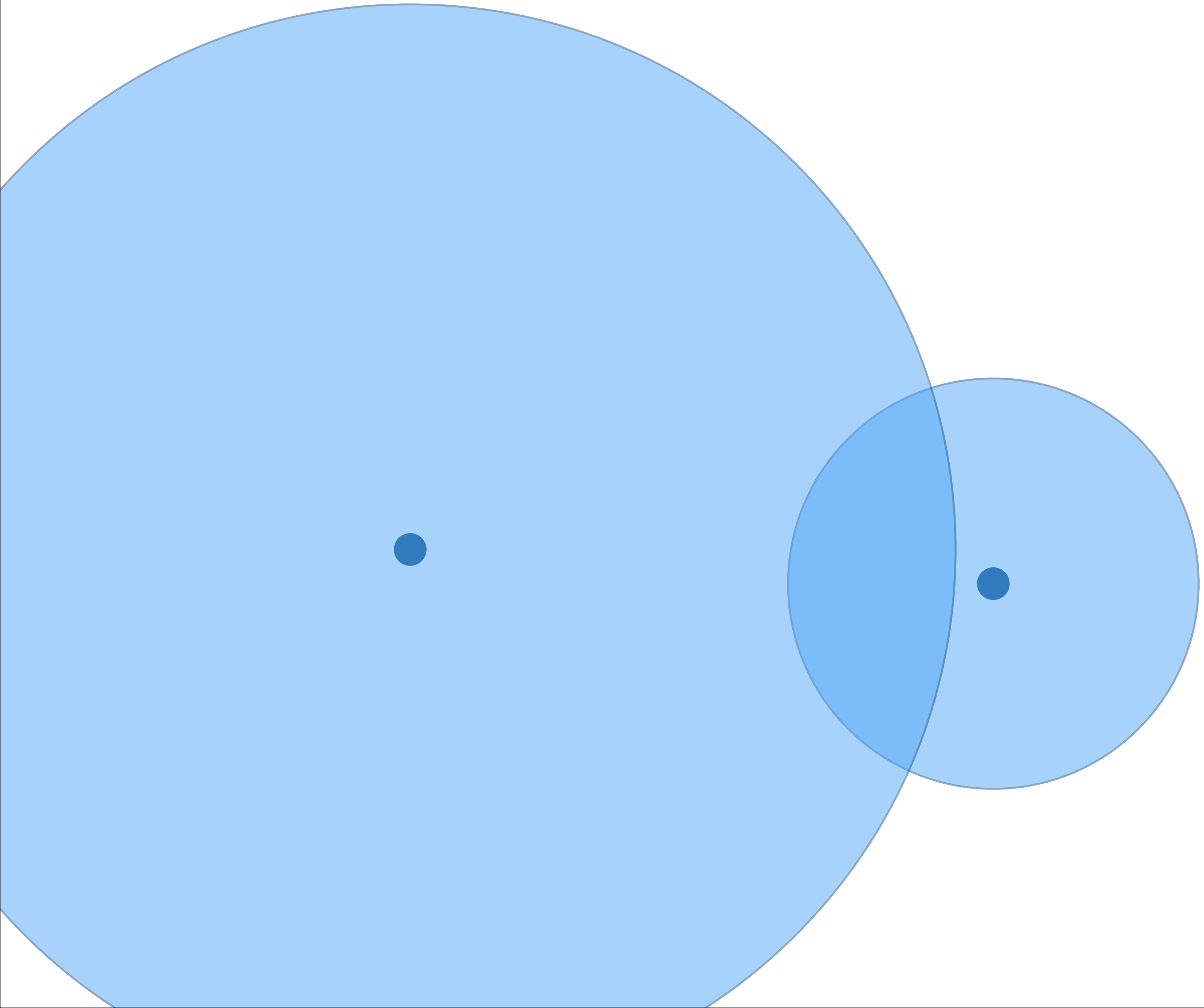
Distance calculation

Distance calculation

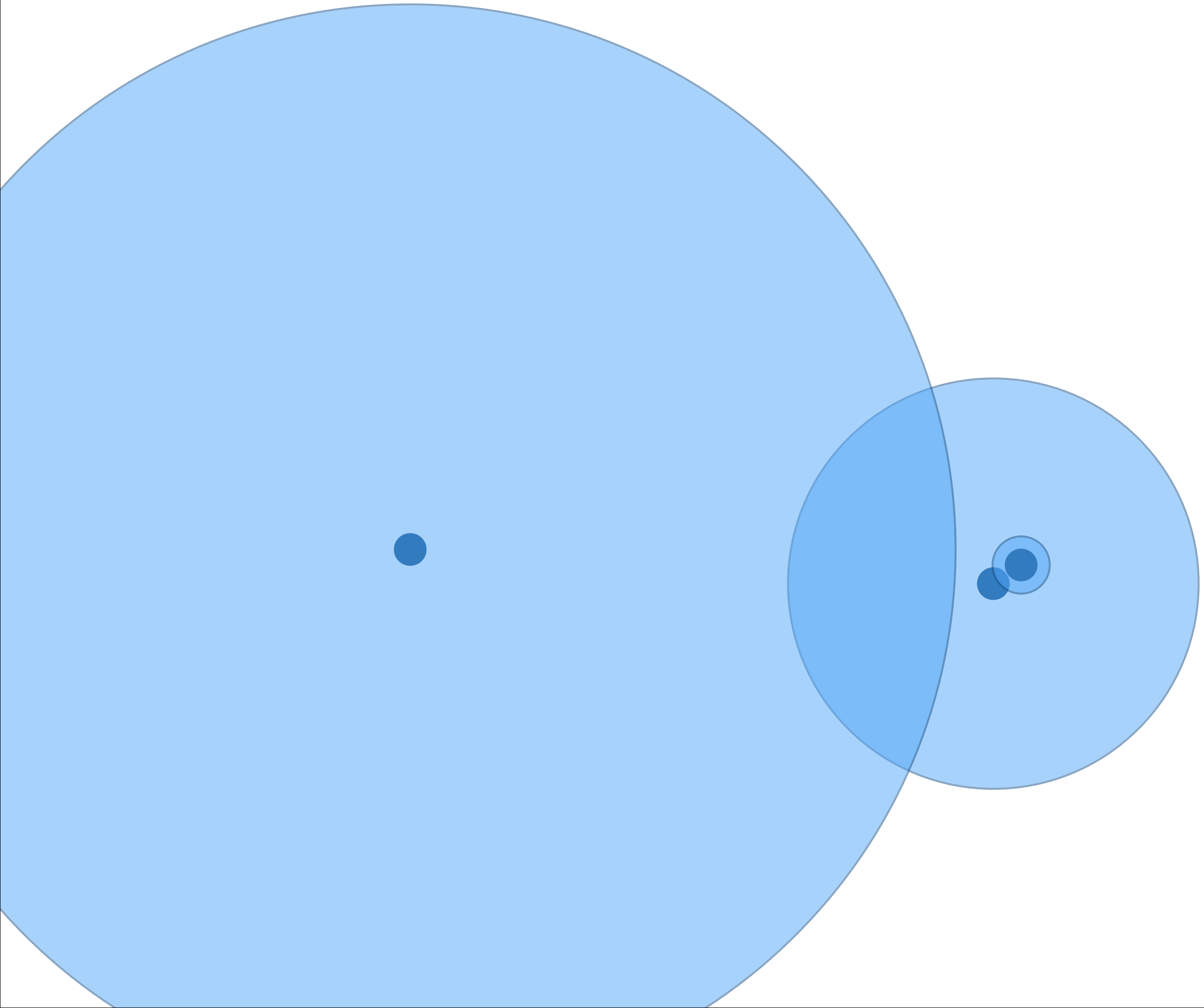
Distance calculation



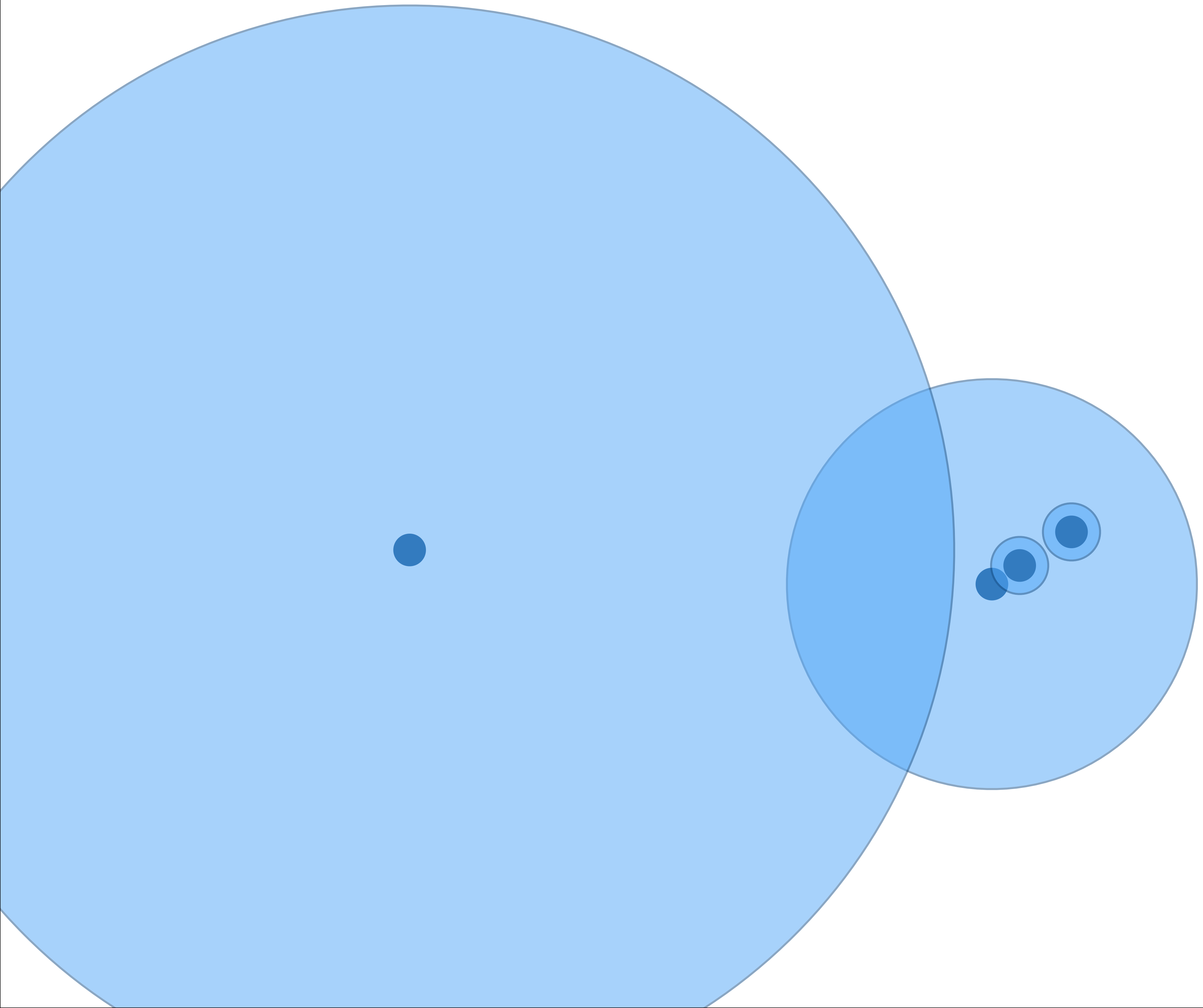
Distance calculation



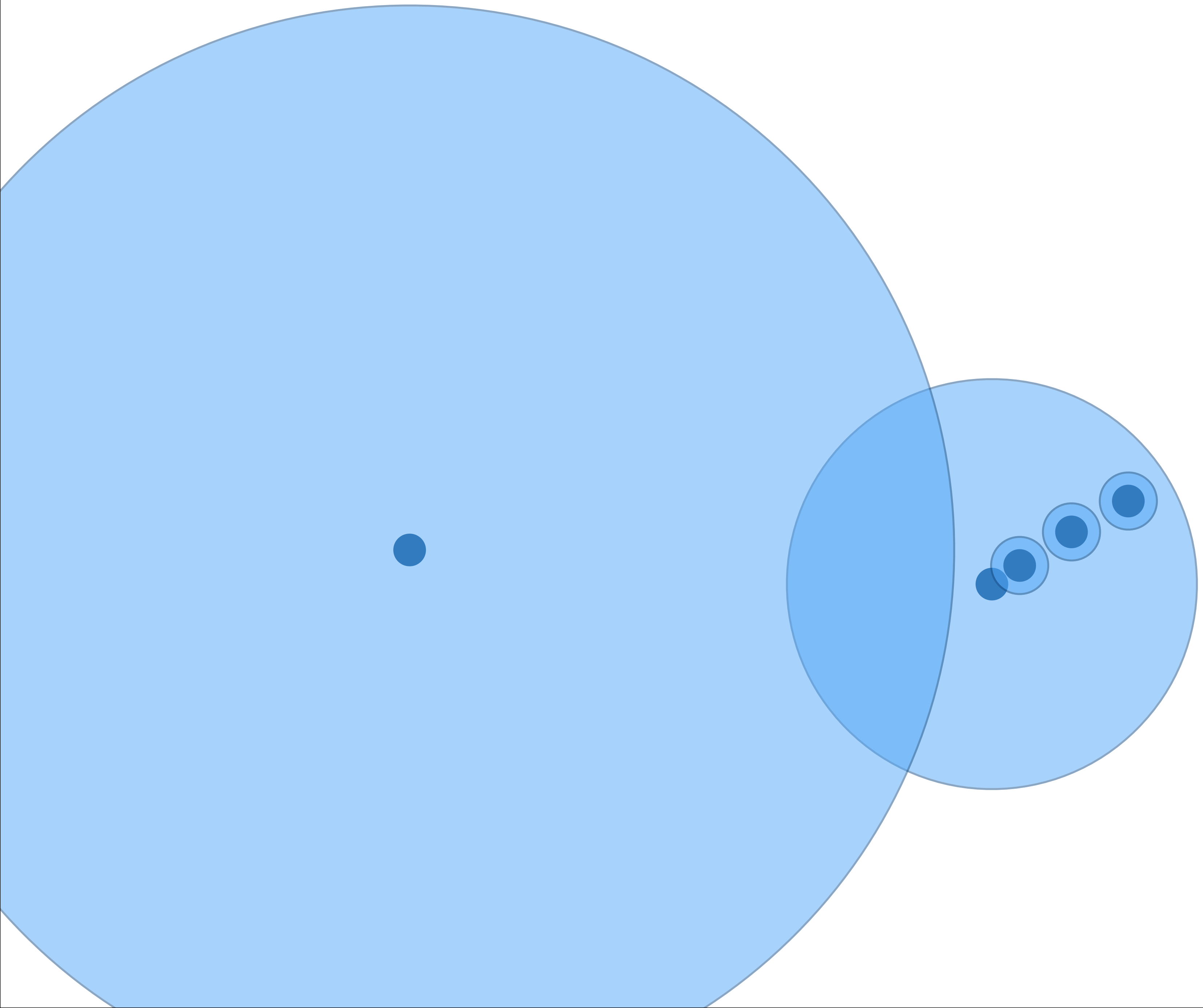
Distance calculation



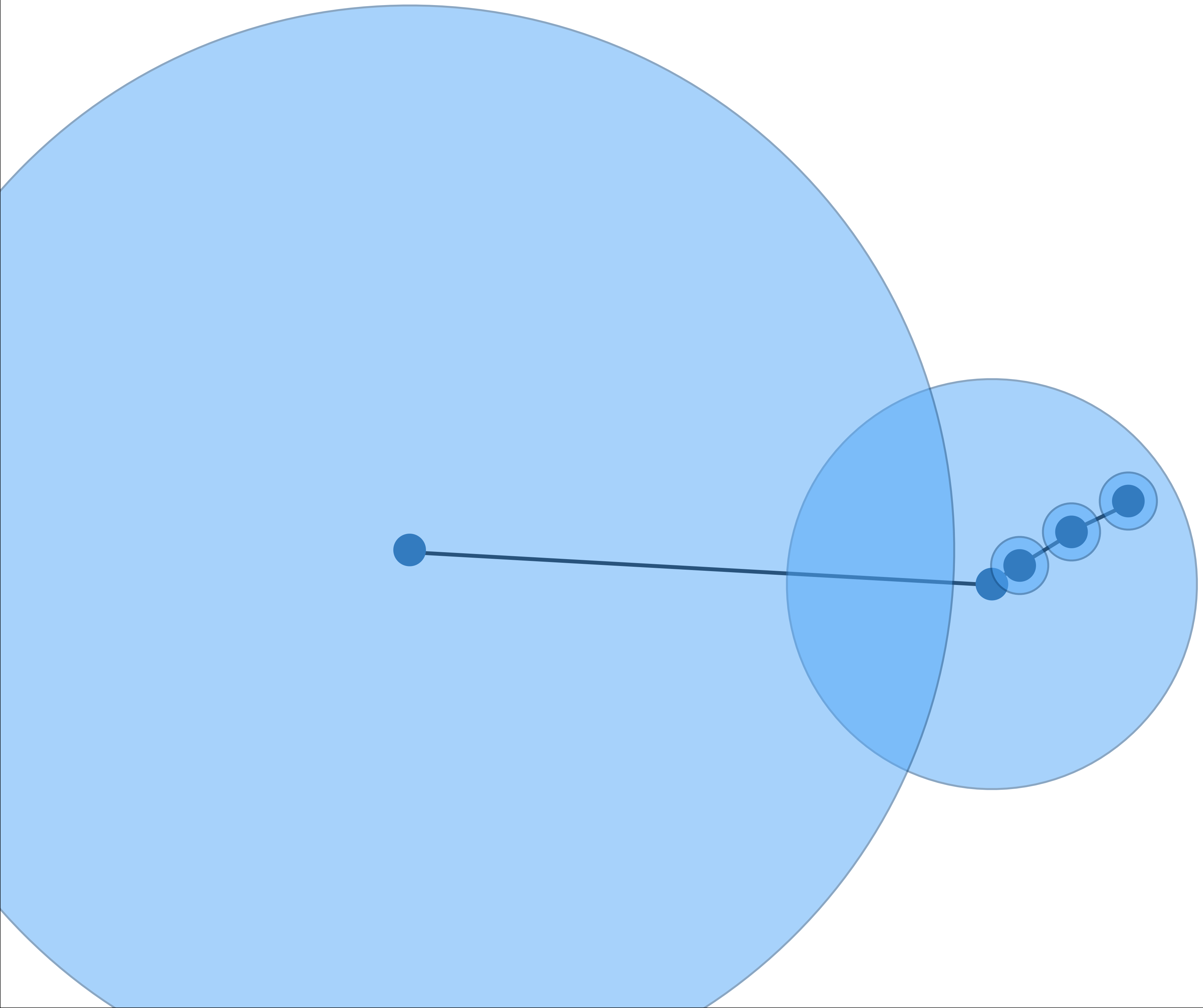
Distance calculation



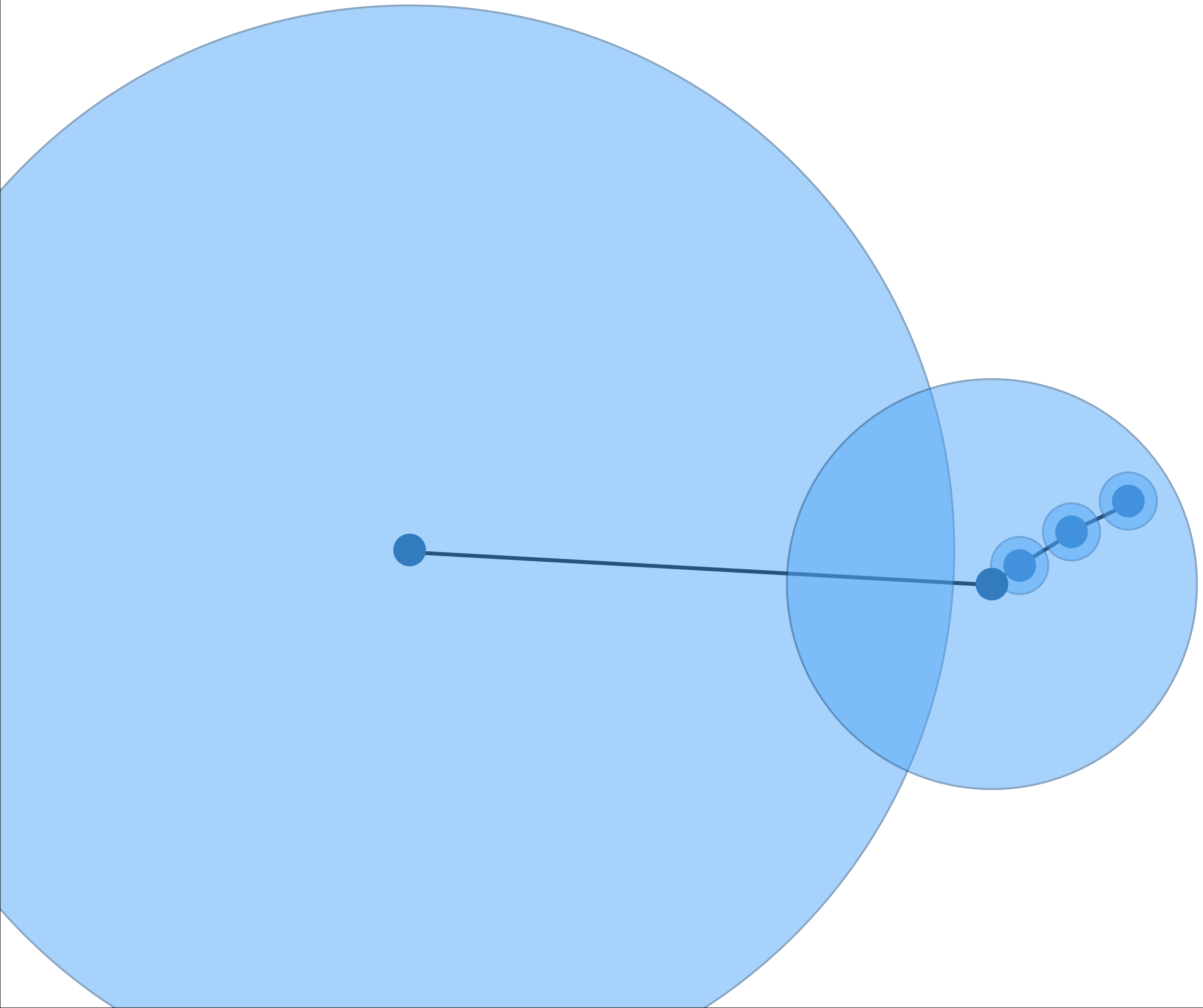
Distance calculation



Distance calculation

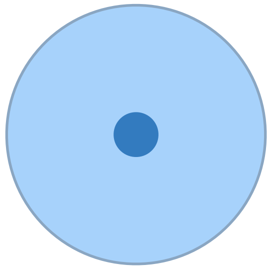


Distance calculation

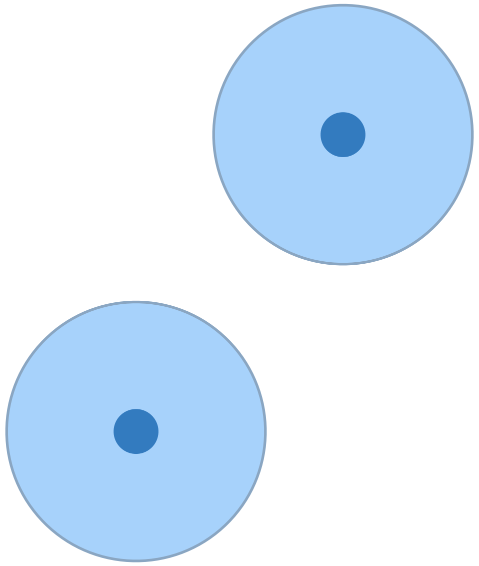


Distance calculation

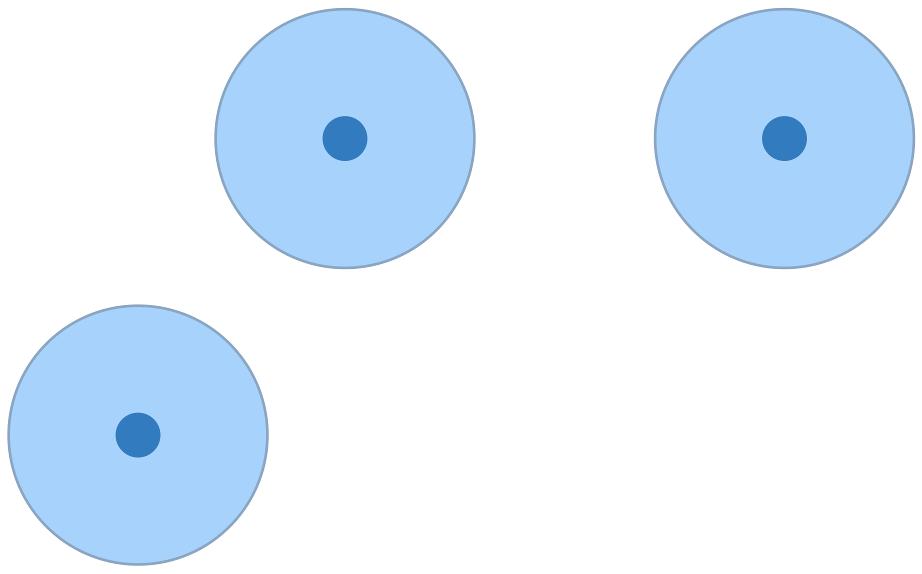
Distance calculation



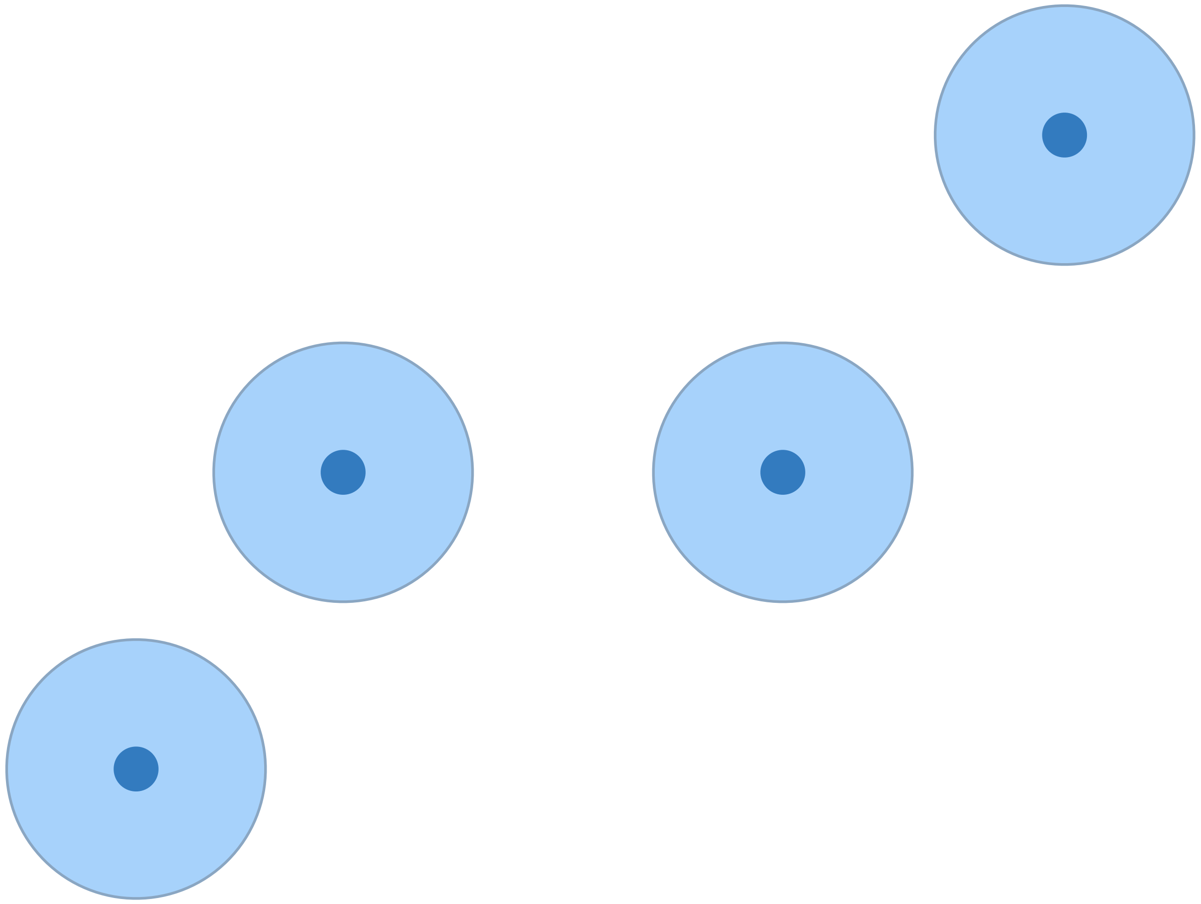
Distance calculation



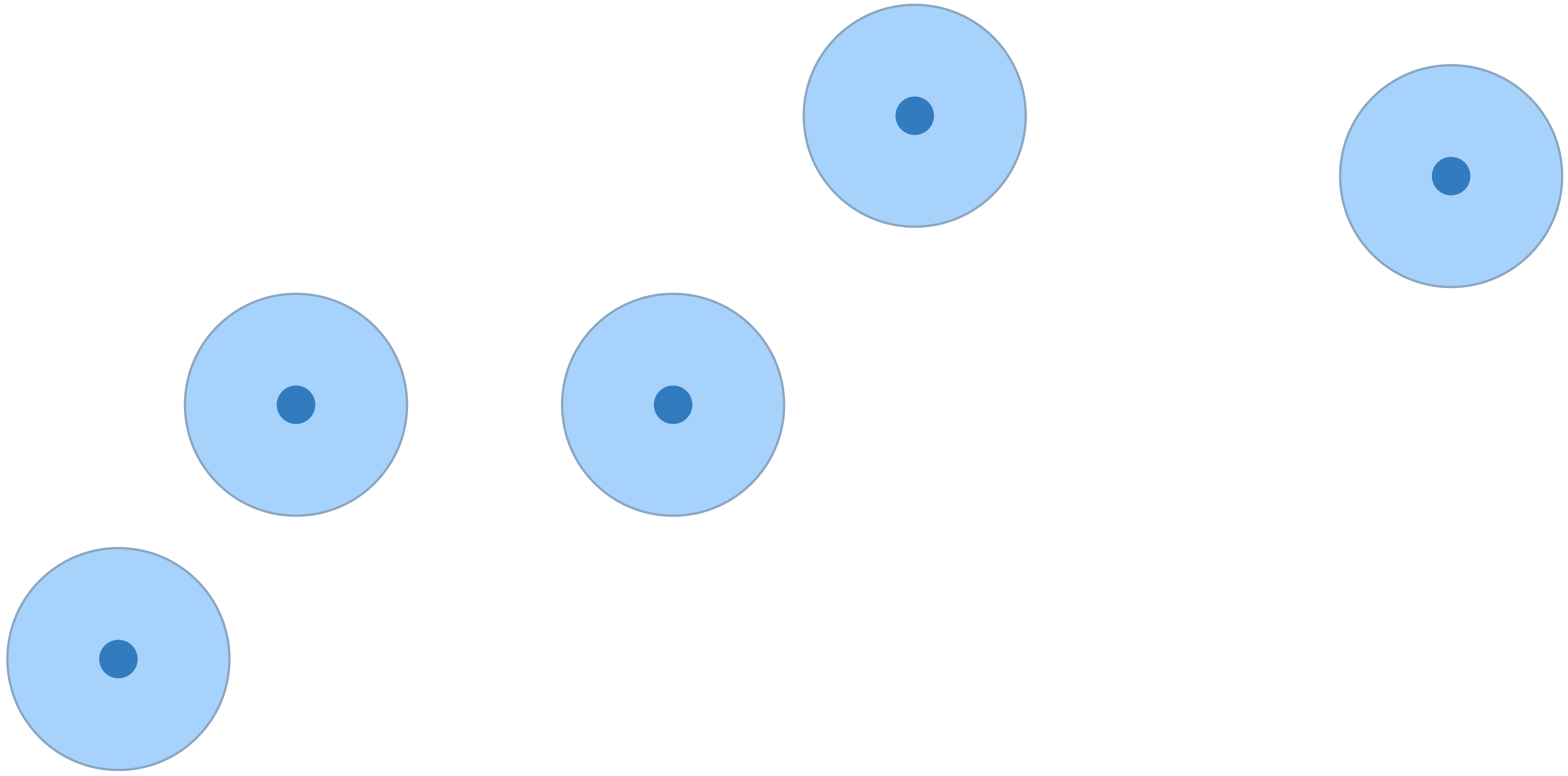
Distance calculation



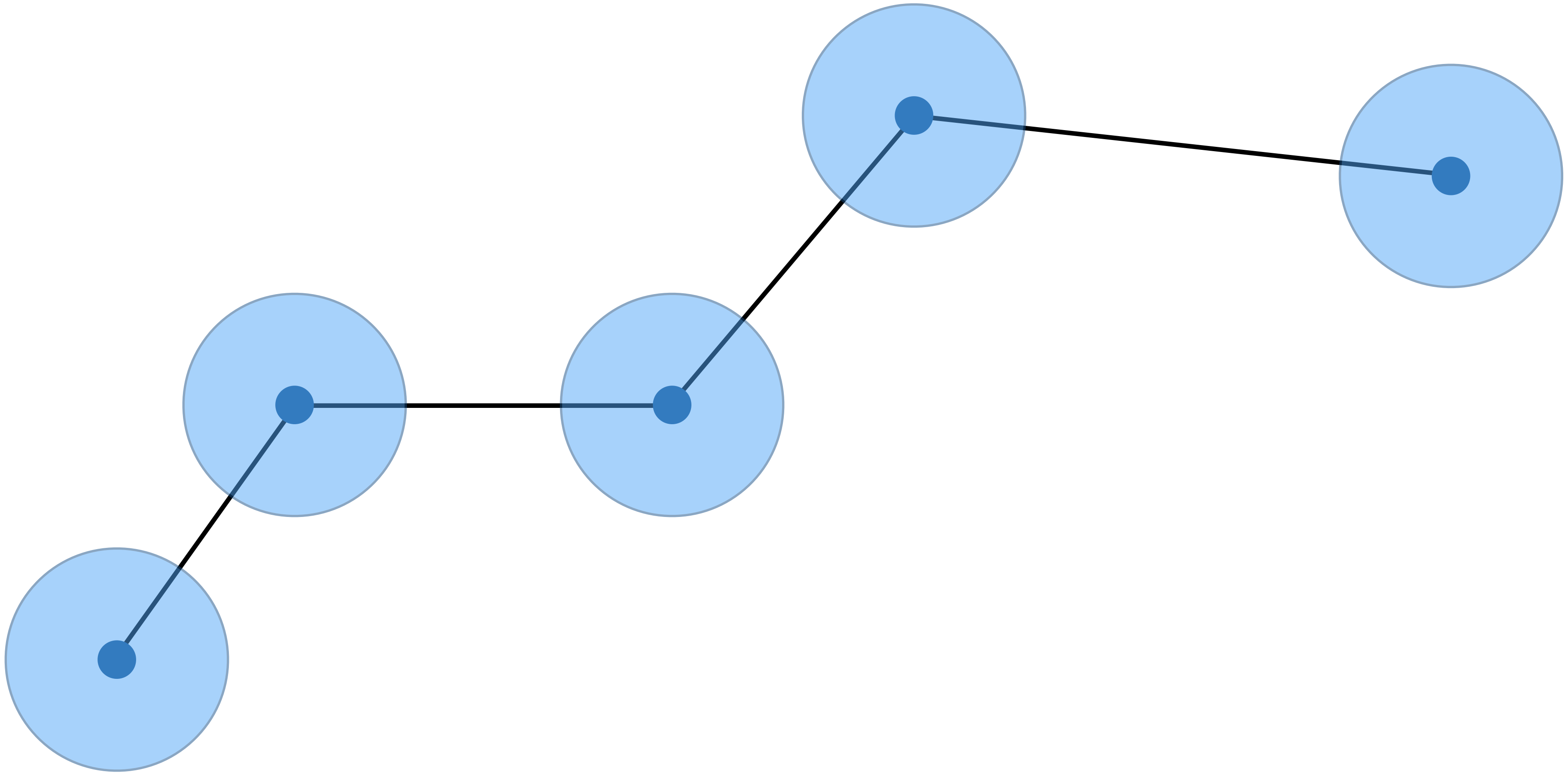
Distance calculation



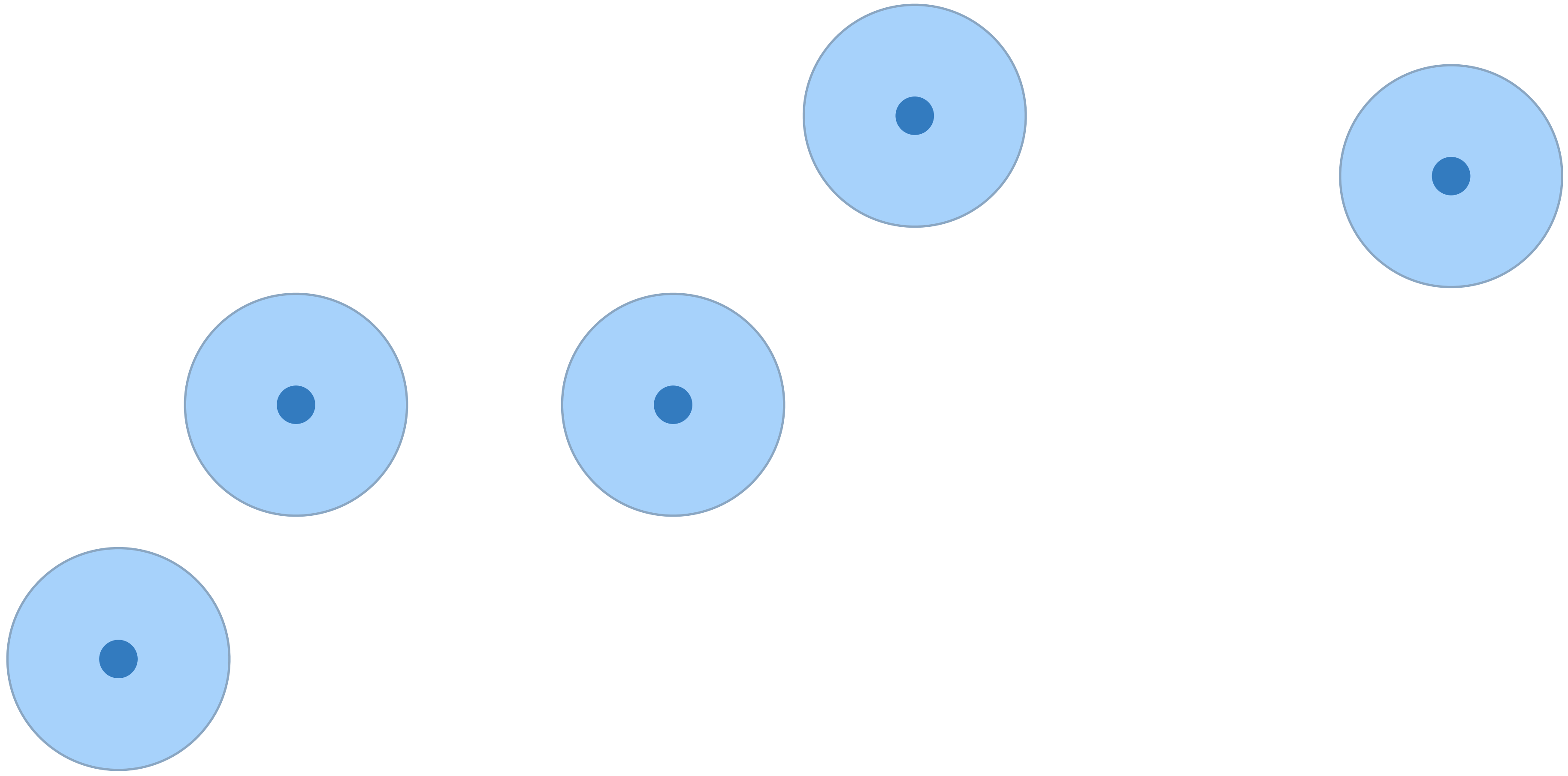
Distance calculation



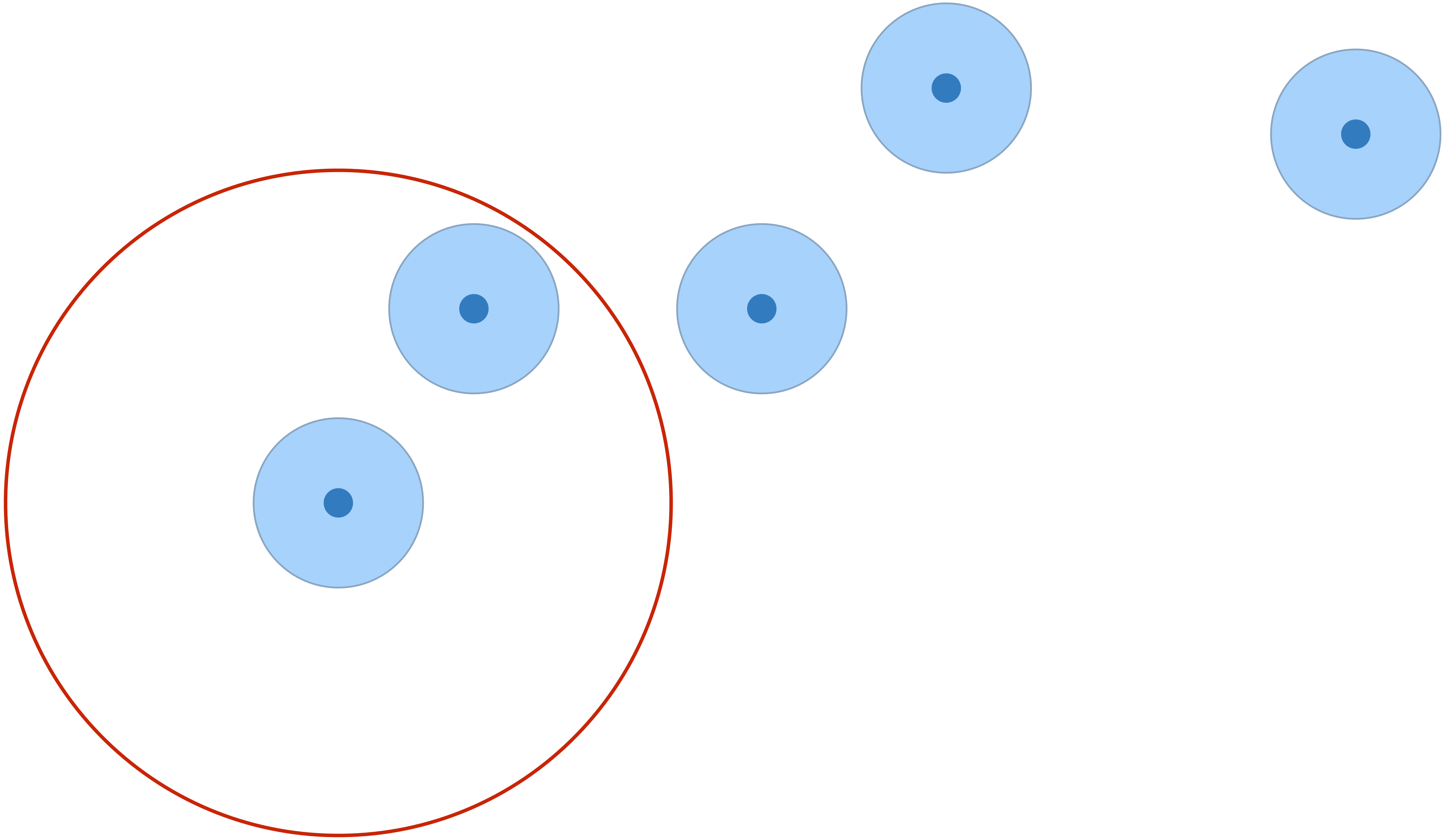
Distance calculation



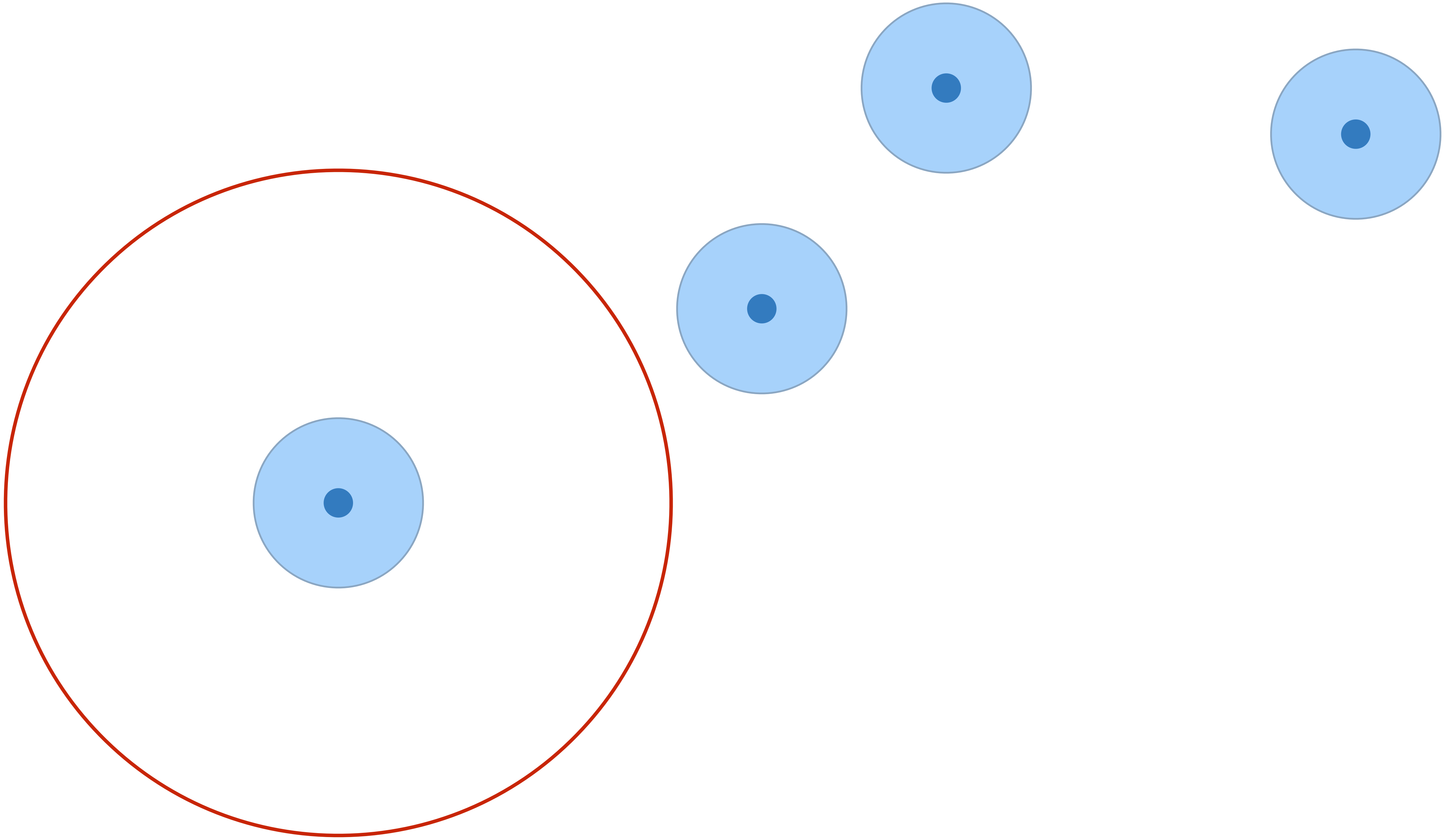
Distance calculation



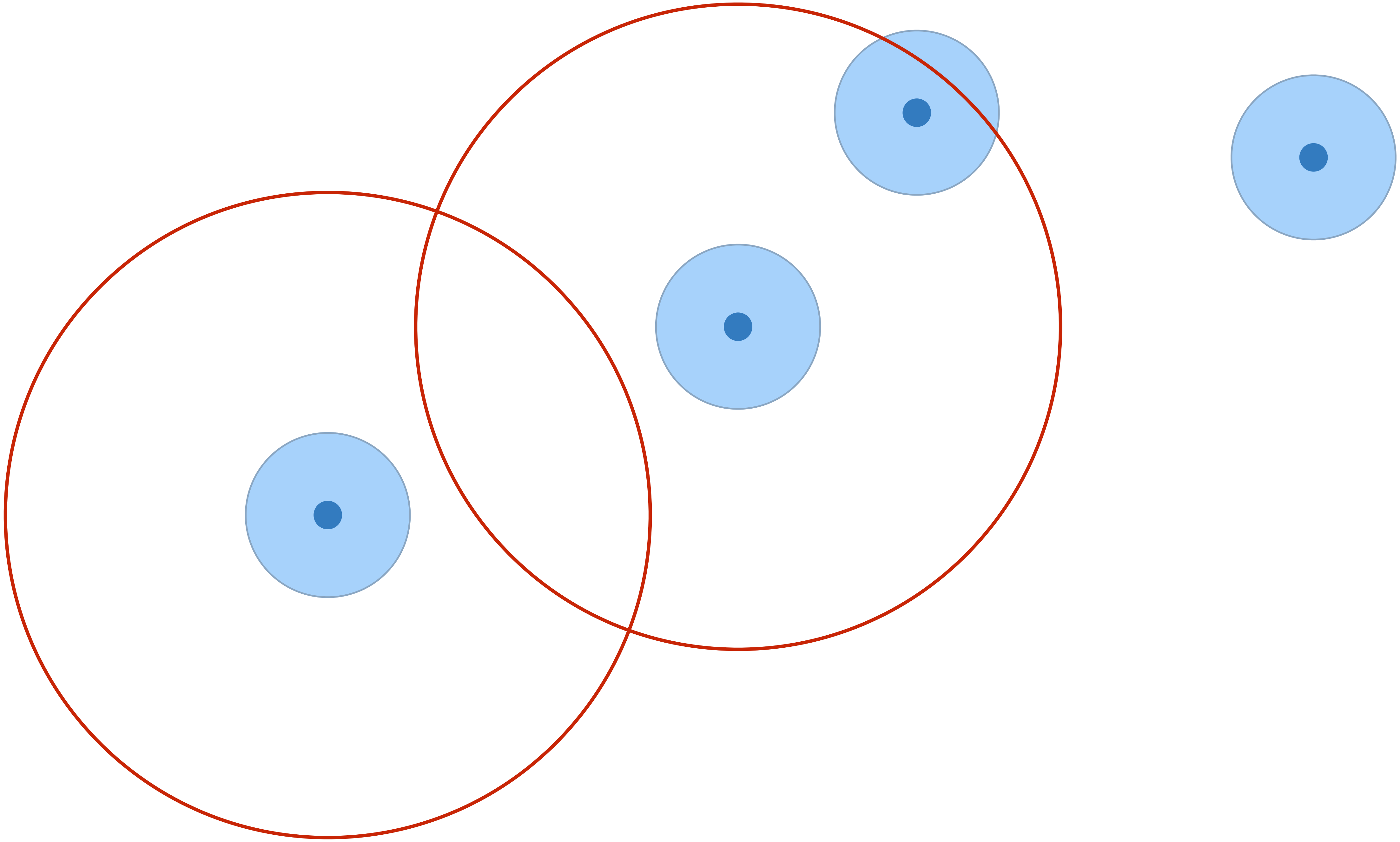
Distance calculation



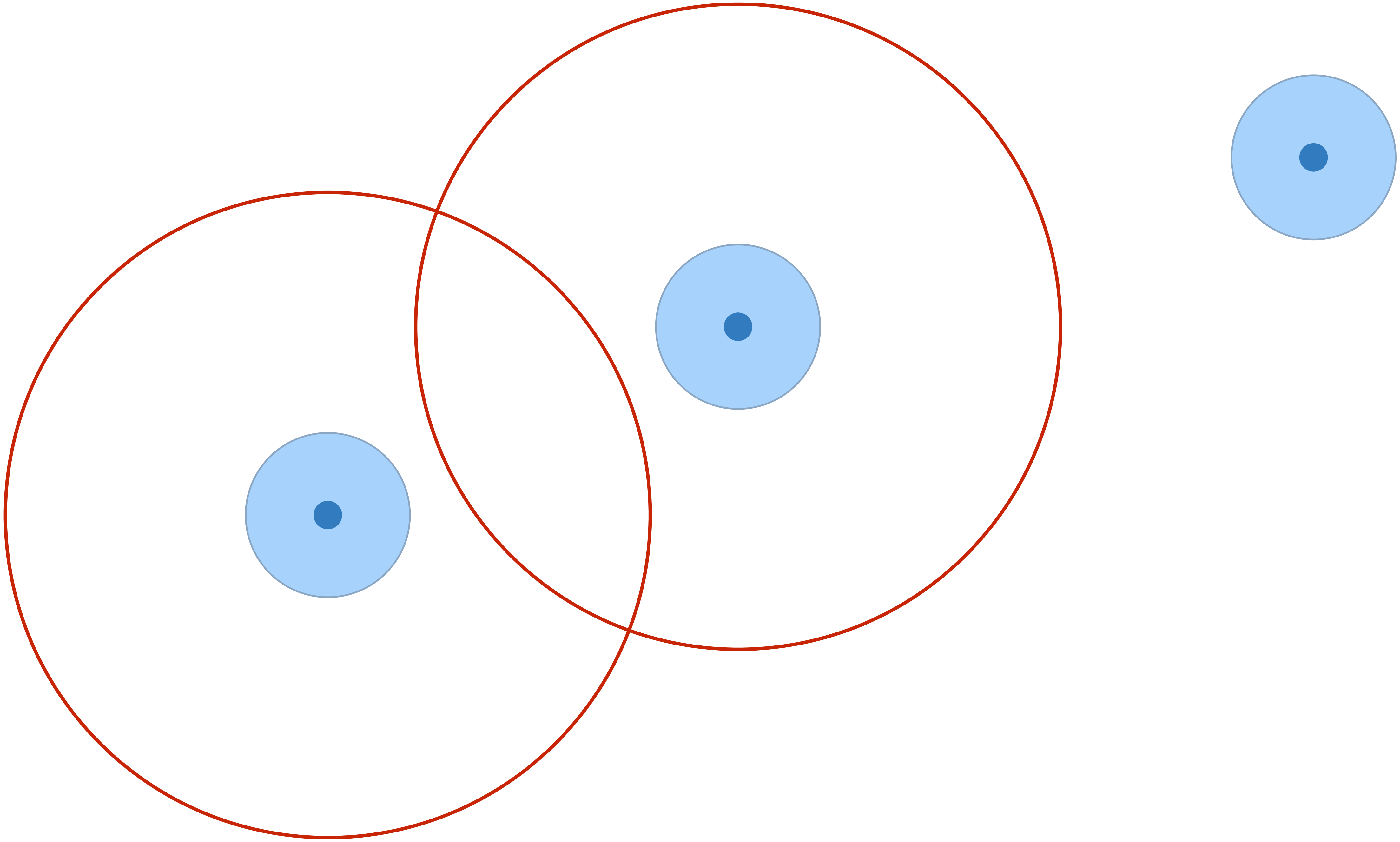
Distance calculation



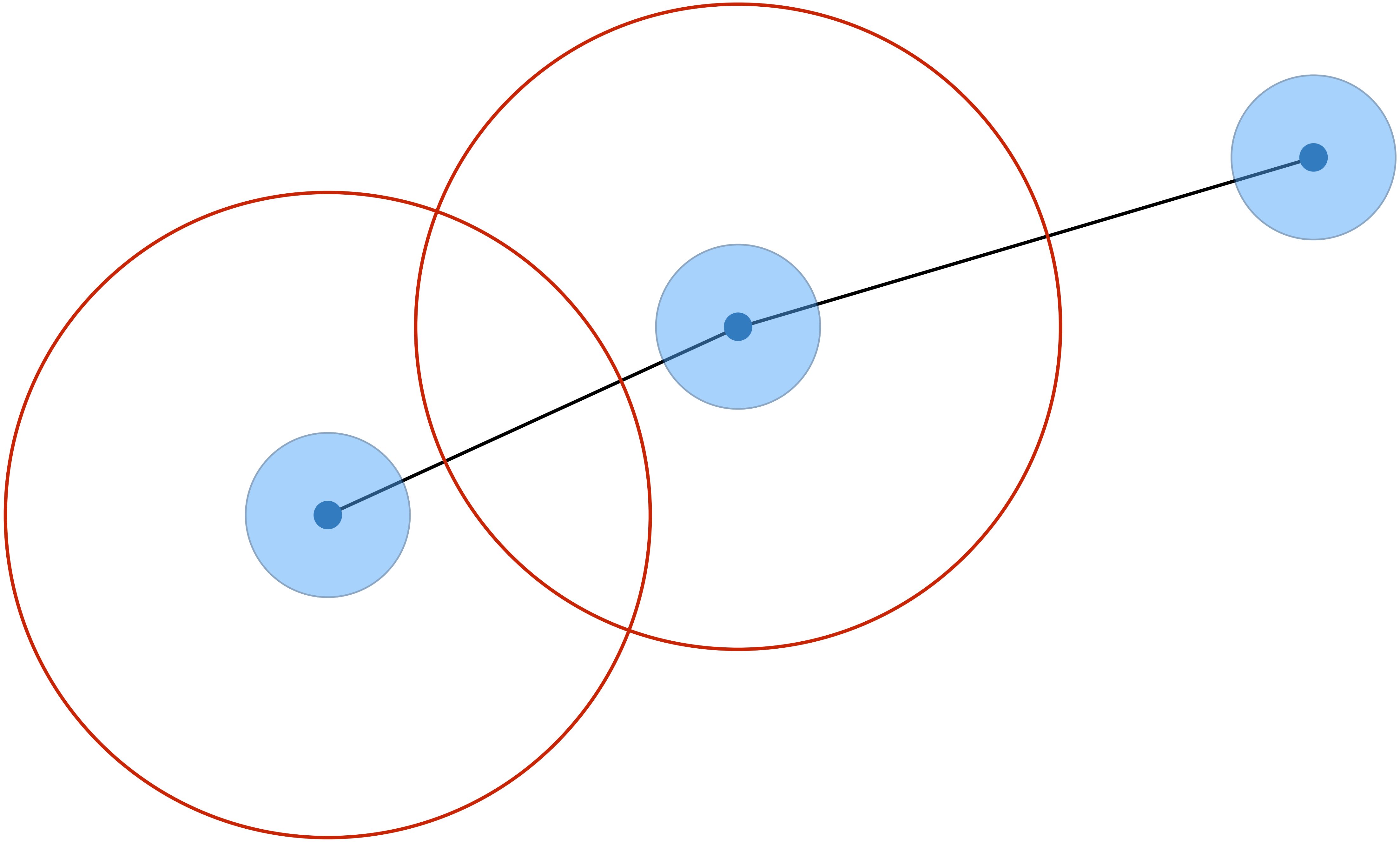
Distance calculation



Distance calculation



Distance calculation



Distance calculation

Remove points that

1. have unacceptably low accuracy
2. are too close to the points before or after them

Don’t remove points that

1. the user has seen
2. are “too far”, unless you really know what you’re doing

Distance calculation

Location data point’s “accuracy”

is not a scientific error value.

Distance calculation

The Earth is not flat!

Don’t do cartesian maths.

Thank you!

QCon

Please evaluate my talk via the mobile app!