

Sitecore xDB in Depth: Harnessing Contacts, Sessions and Clusters

Dmytro Shevchenko

May 18, 2017

#sugcon



SUGCON
EUROPE 2017



Some background:

- What is Sitecore xDB?
- How does it scale?

Fun stuff:

- How does xDB deal with parallelism?
- Practical scenarios of working with contact APIs.

From a brand's perspective:

- Collect all available customer data;
- Gain insights via reporting;
- Enhance customer experience;
- Profit.

From a developer's perspective, xDB looks like this:

- Backend user interfaces;
- Data storage;
- Data and behavior APIs;
- Server infrastructure.

Scaling the infrastructure

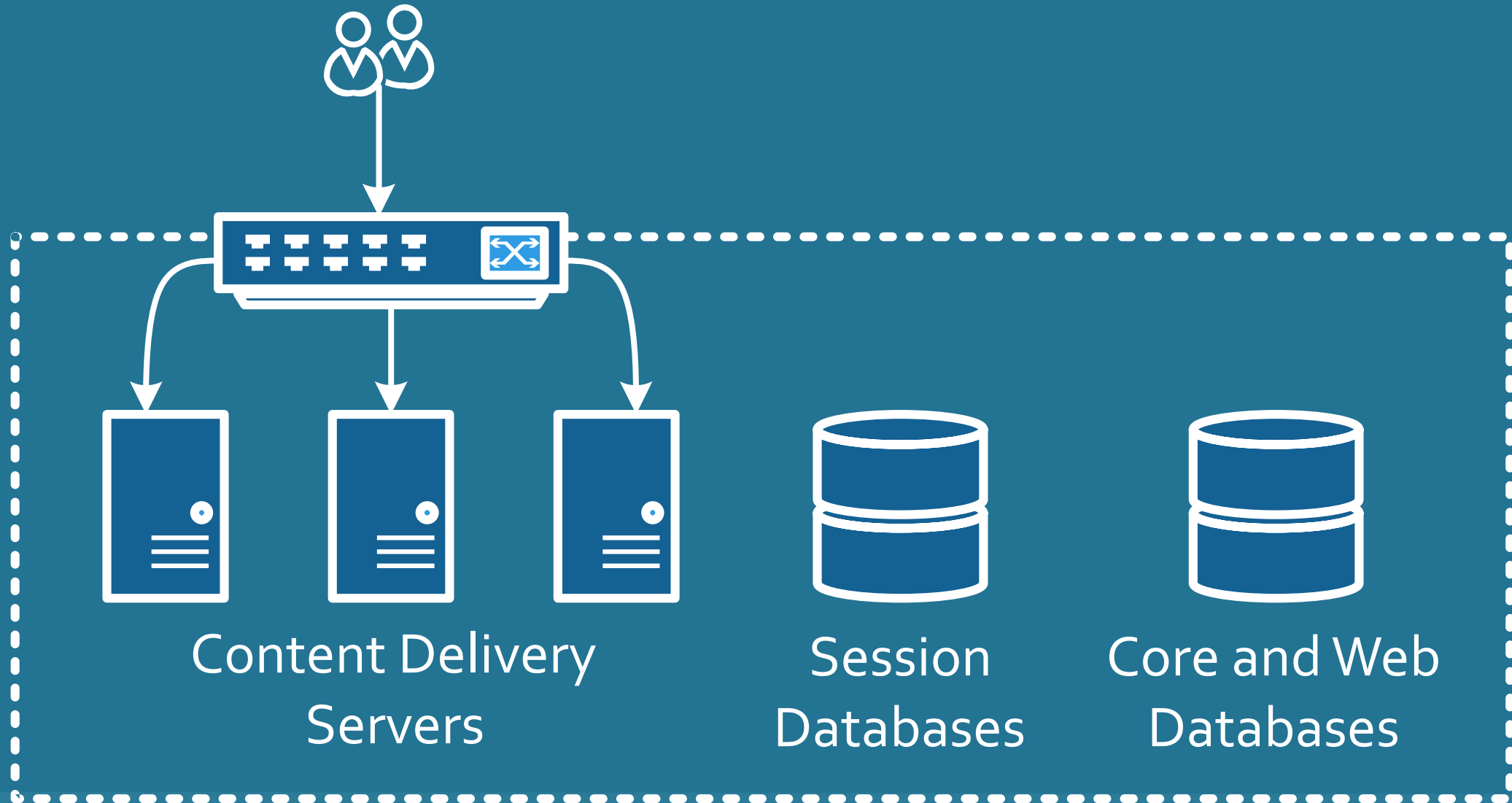
The image features a teal background on the left side. The text "Scaling the infrastructure" is written in white, bold, sans-serif font. On the right side, there are several overlapping diagonal stripes in shades of purple and orange, creating a modern, geometric design.



xDB was built to support:

- Vertical scaling;
- Horizontal scaling;
- Geographically distributed solutions.

Content Delivery Cluster



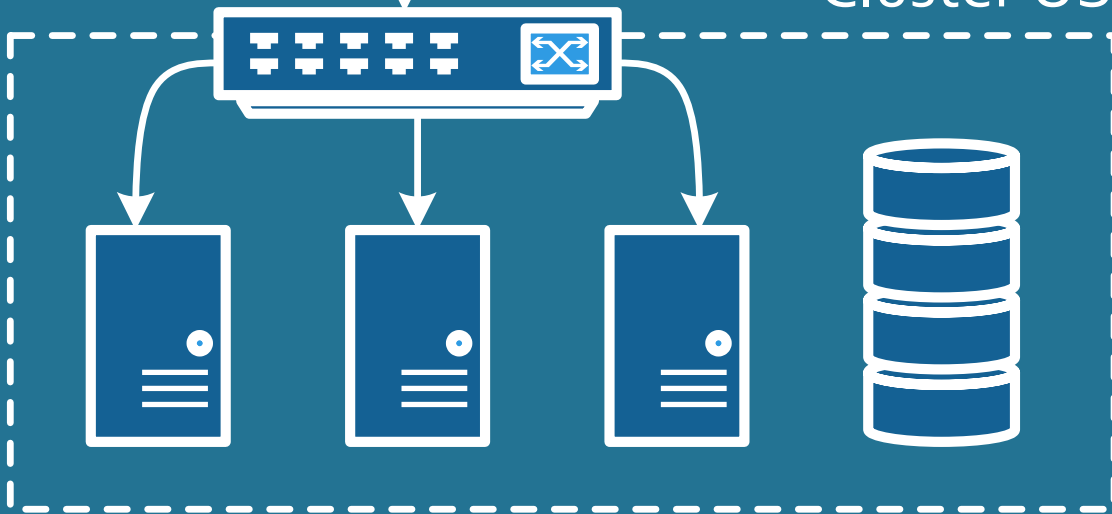
Geographically distributed setup

 American visitors

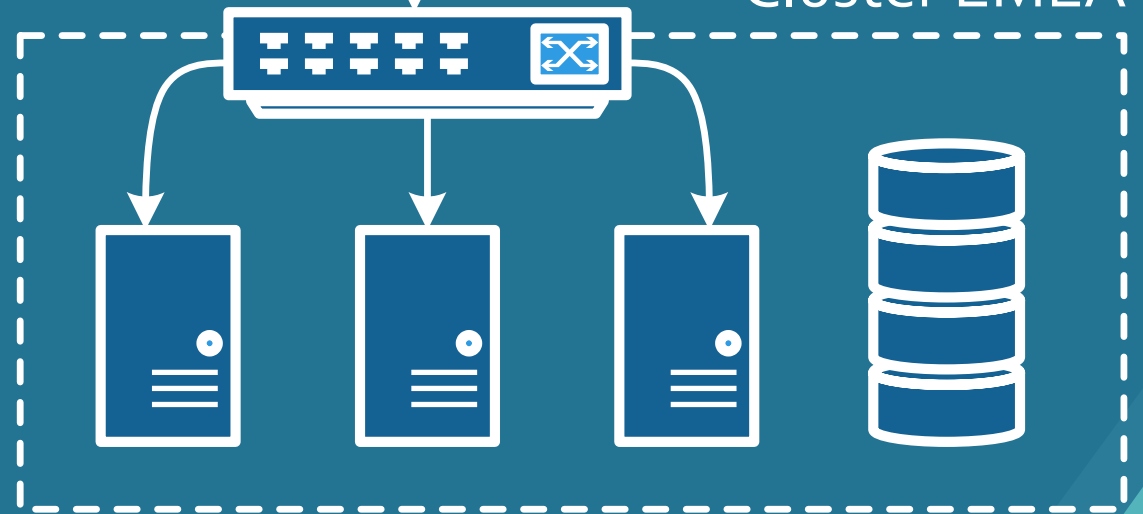
 European visitors

Geo-aware DNS

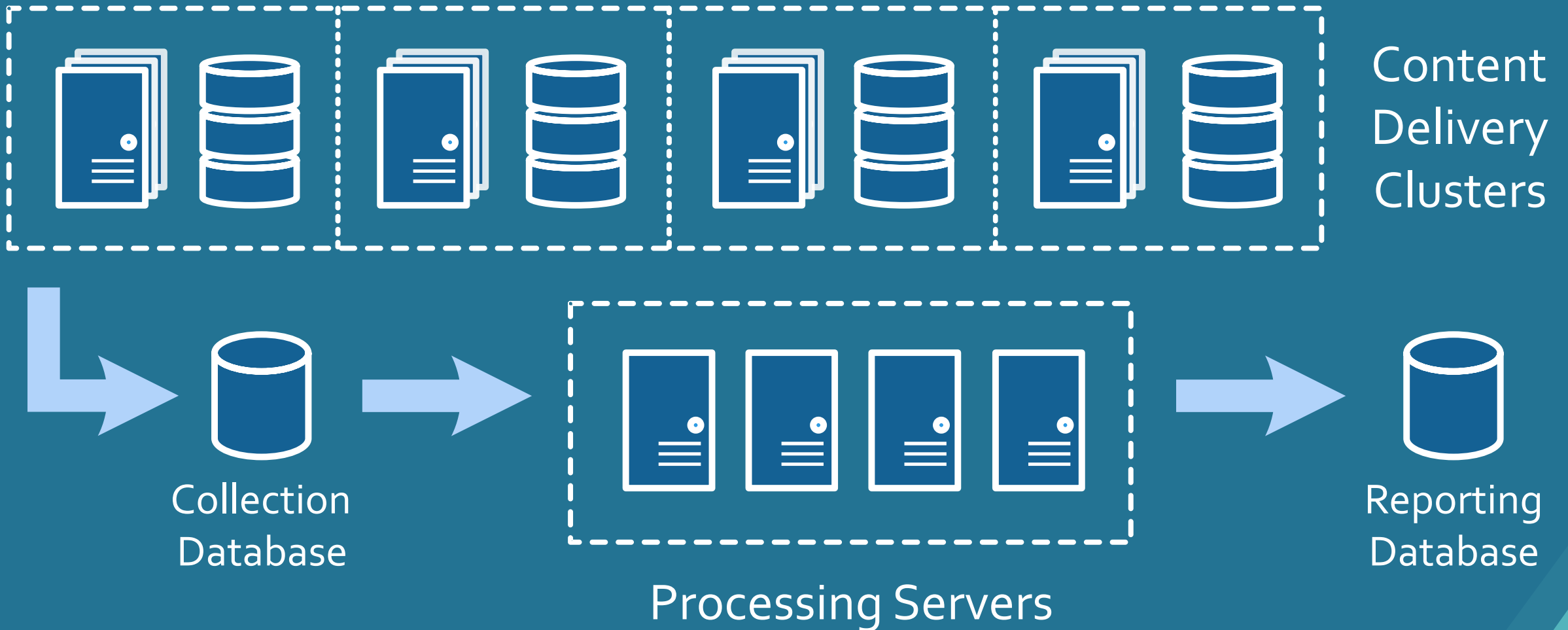
Cluster US



Cluster EMEA



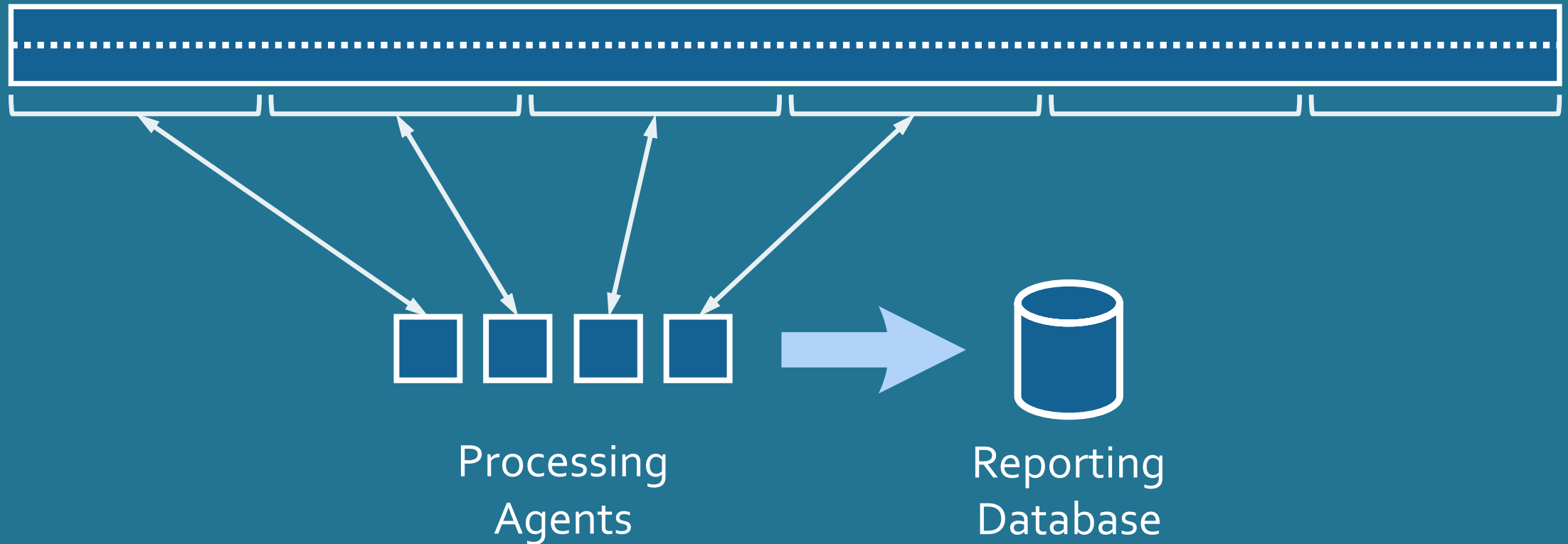
Processing distribution



xDB is a complex development environment.
It scales across many servers and
heavily relies on parallelism.

Concurrency in data processing

Processing pool of interactions



xDB Contacts and Concurrency Control



What is a contact?

- Representation of a real person;
- Can have multiple interactions;
- Stored as a document in the Collection database.

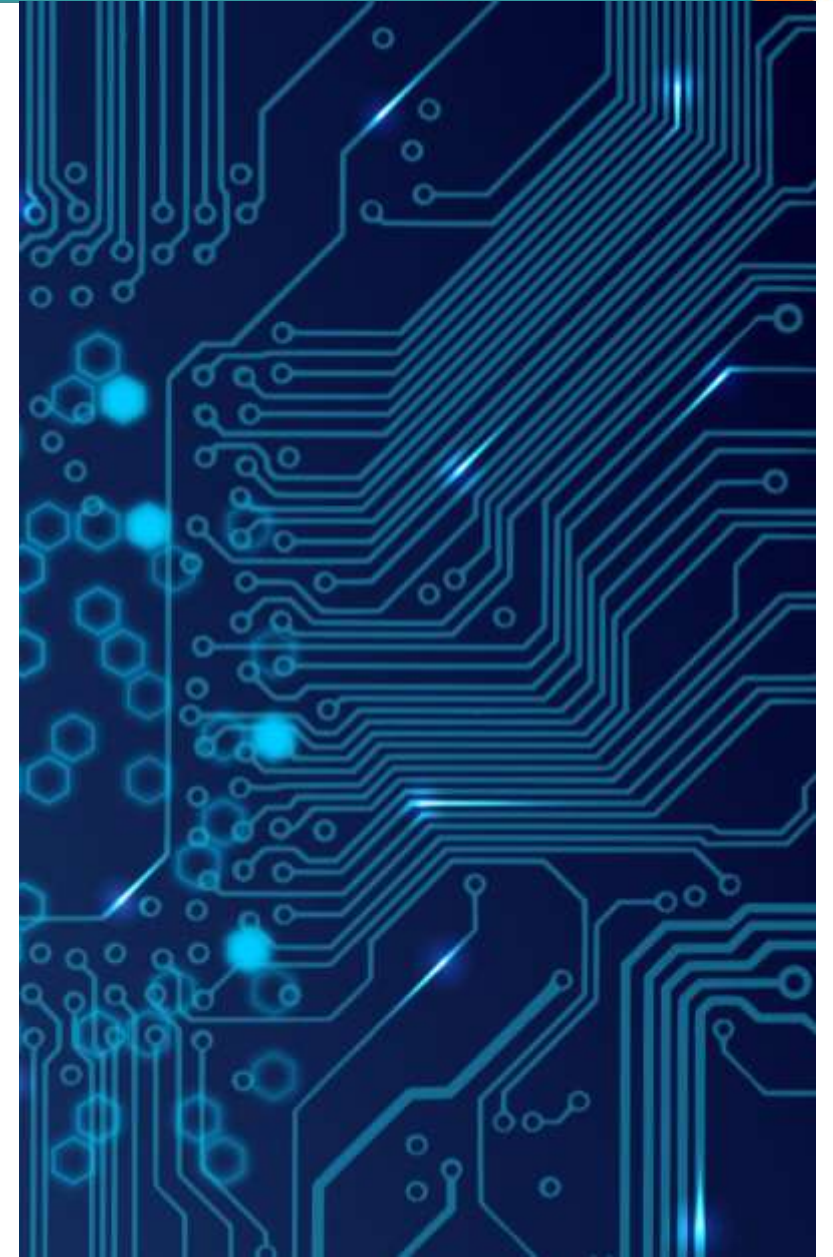


Sample scenarios:

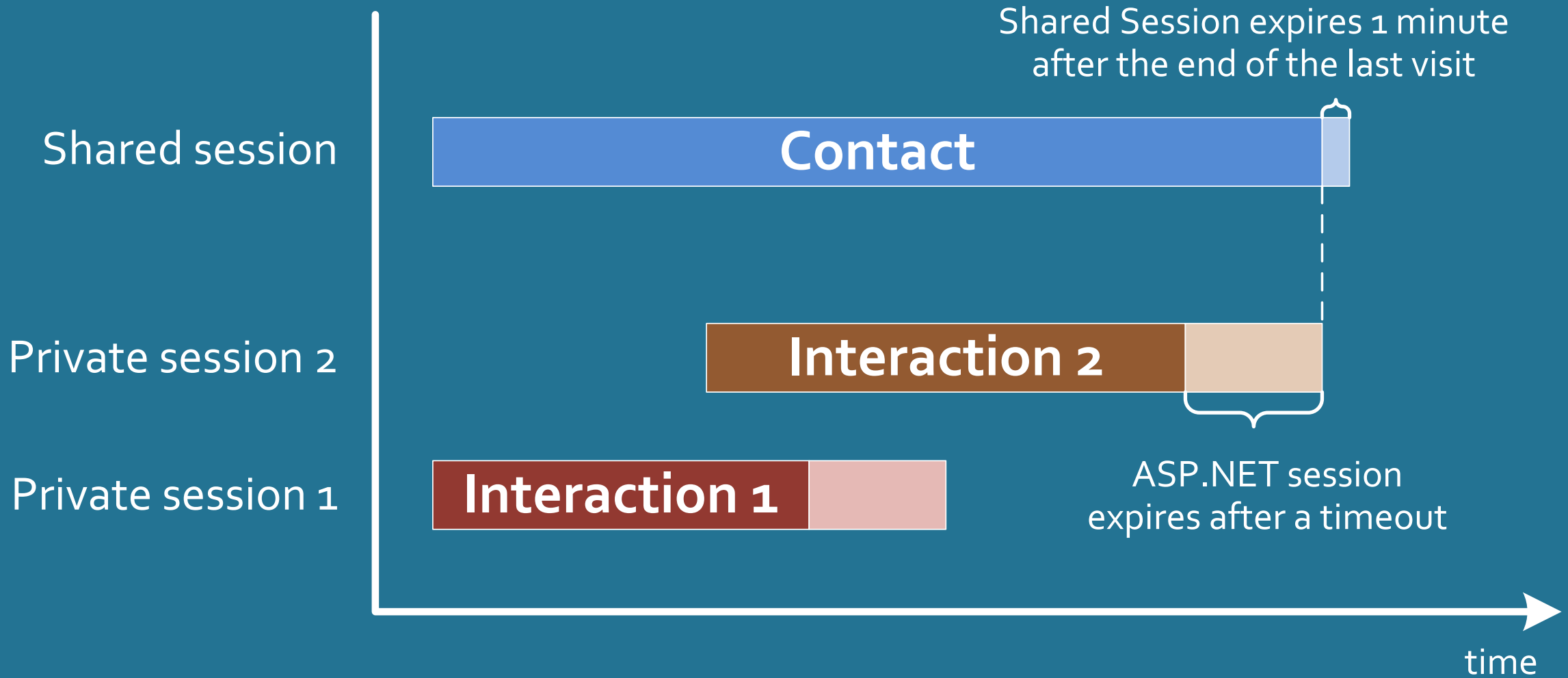
1. A page where a user can update their personal info
2. Contact data import that runs in the background
3. An administrative page that allows to modify contacts

There are four sets of APIs:

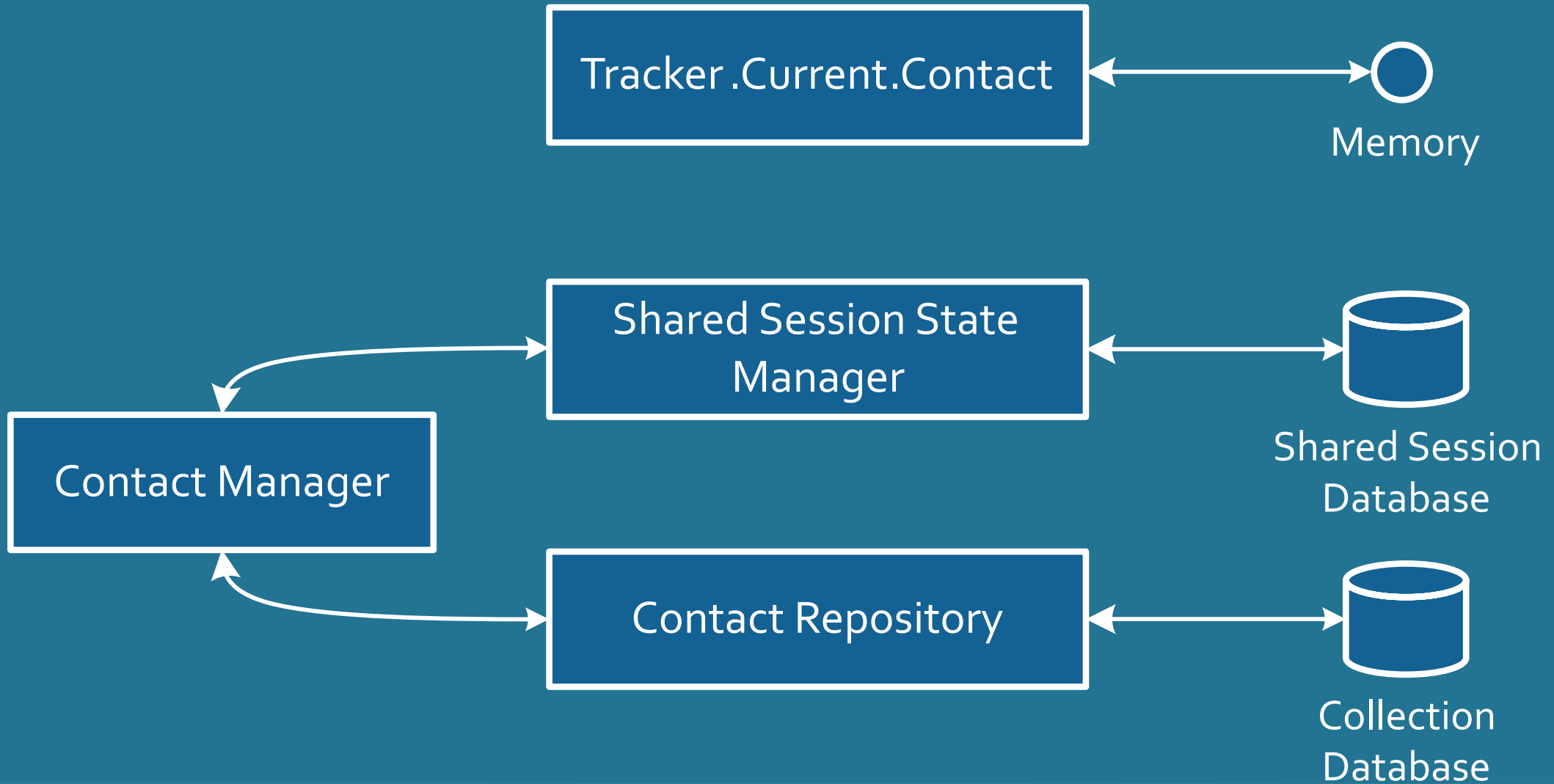
- ContactRepository
- ContactManager
- SharedSessionStateManager
- Tracker.Current.Contact



Session lifetime



Three types of contact storage



Don't know which API to use?



Contact lock stages

Tracker



Shared Session
Database



Collection
Database

In custom code, use the storage API that corresponds to the level of your code

A page where any user can update their own personal info

- Easy: just use `Tracker.Current.Contact`


Contact data import that runs in the background

- Use ContactRepository
- If a contact is locked, postpone its modification

A page that allows an admin to modify any contact

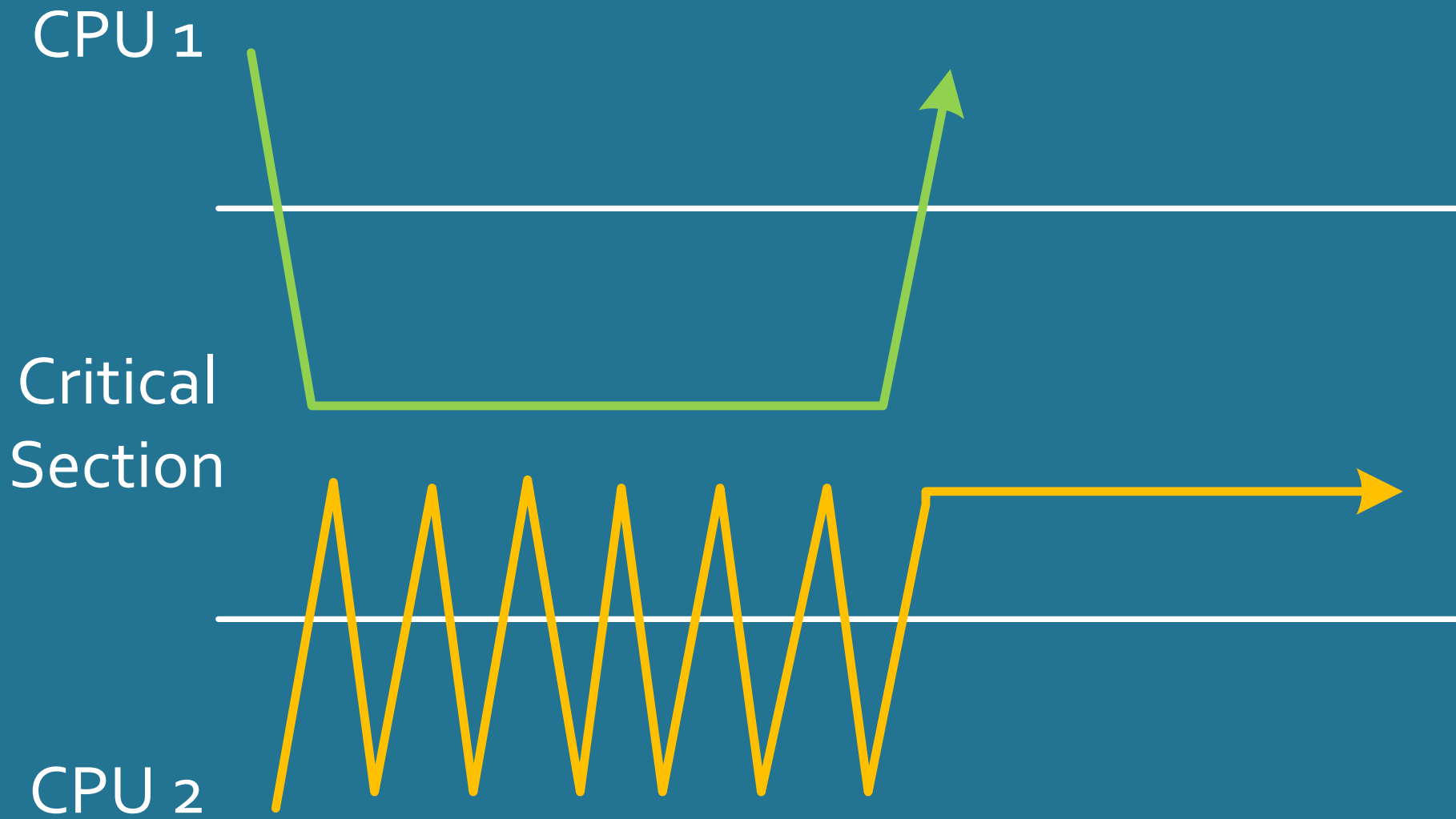
- Try Tracker.Current.Contact
- Try Shared Session
- Try Collection Database
- Save the contact to the storage where you found it

Concurrency Control: Technical Details

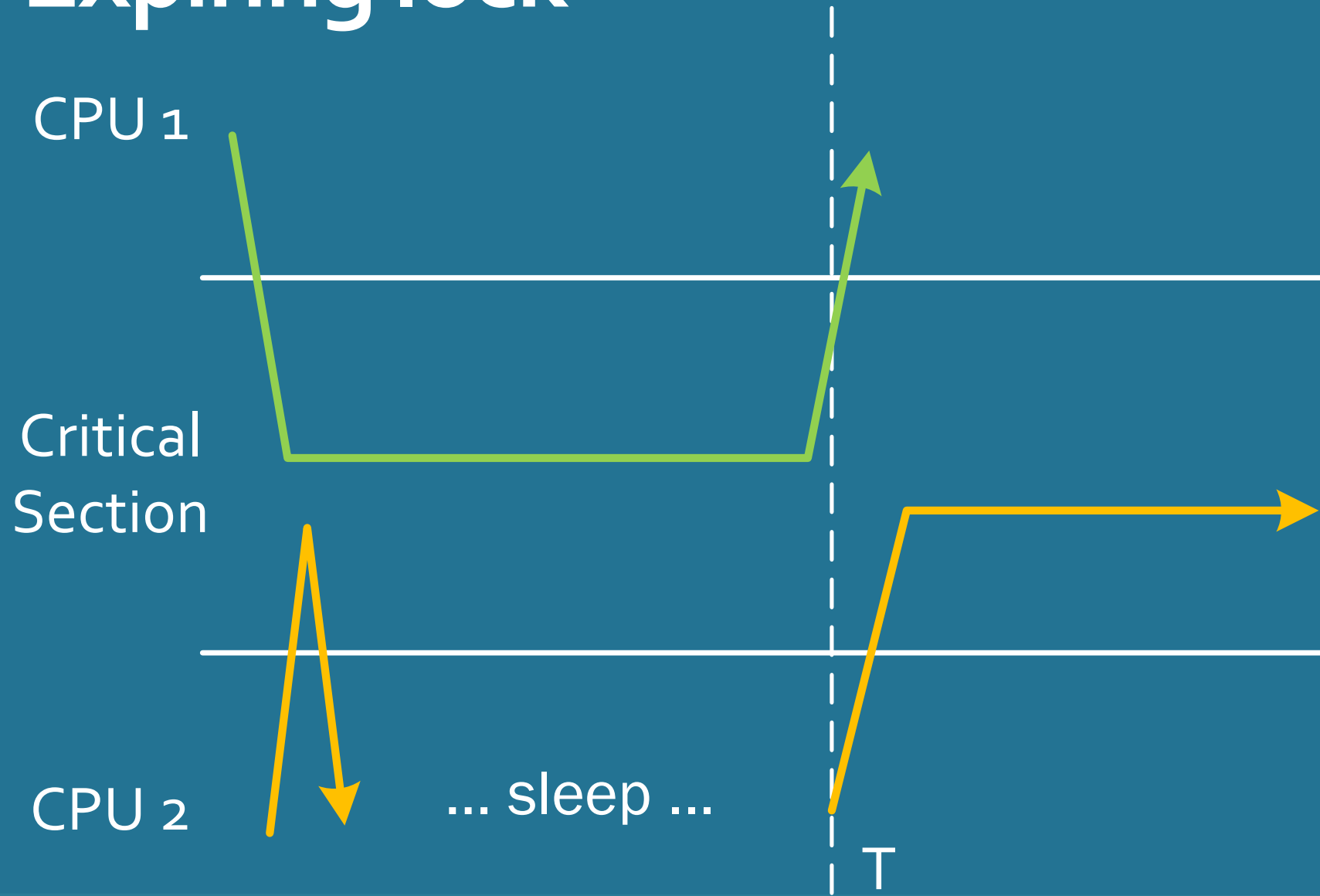


- Critical section (lock/monitor)
- Mutual exclusion (mutex)
- Semaphore
- SpinLock
- WaitHandle
- ...

Spin lock



Expiring lock



Concurrency control examples

guaranteed
memory access

Tracker.Current.Contact

Contact locked
(short-term)



spin-wait

Shared Session State
Manager

Shared Session
Database

Contact locked
until 2017-02-16 15:33:02



postpone

Contact Repository

Collection
Database

Alternative models:

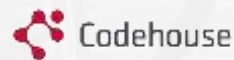
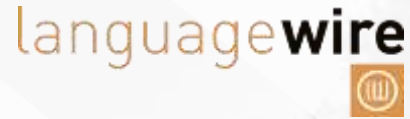
- Optimistic locking
- Event sourcing
- ...

Some rumors:

- Contact APIs will probably be superseded by xConnect
- Locking model may be revised

Take-home points:

- Remember about concurrency
- Use the correct API
- Ask on Slack and Sitecore Stack Exchange
- Look forward to xConnect



Thank you!

You can find me online:

@dmytro on Slack

@shedal on Twitter