SUZUKI–KASAMI'S BROADCAST ALGORITHM

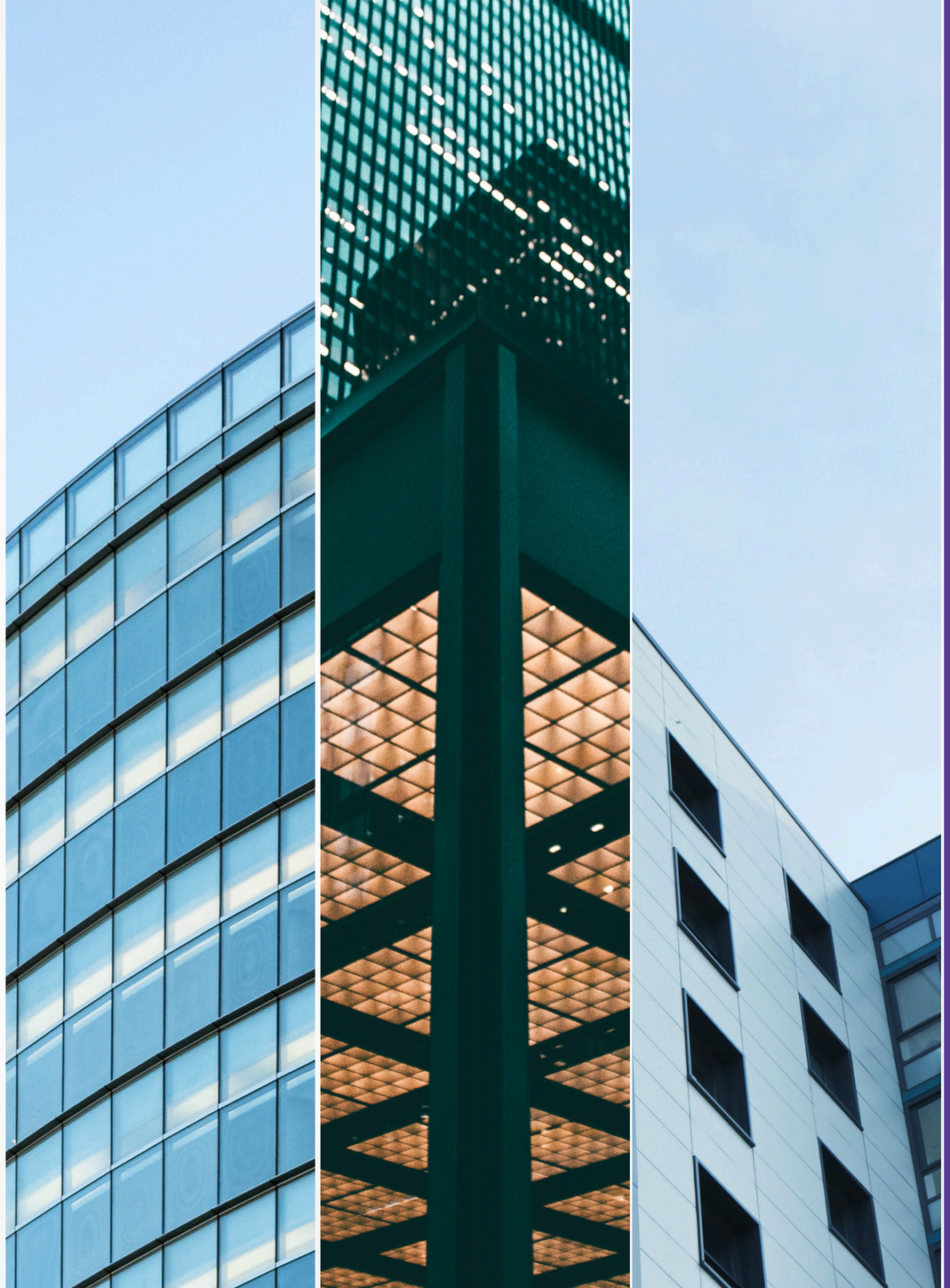
KEY IDEA

Only one node can enter the critical section at a time.

Access is controlled using a single token.

Nodes broadcast requests when they need the token.

Whoever has the token can enter the critical section.



CRITICAL SECTION

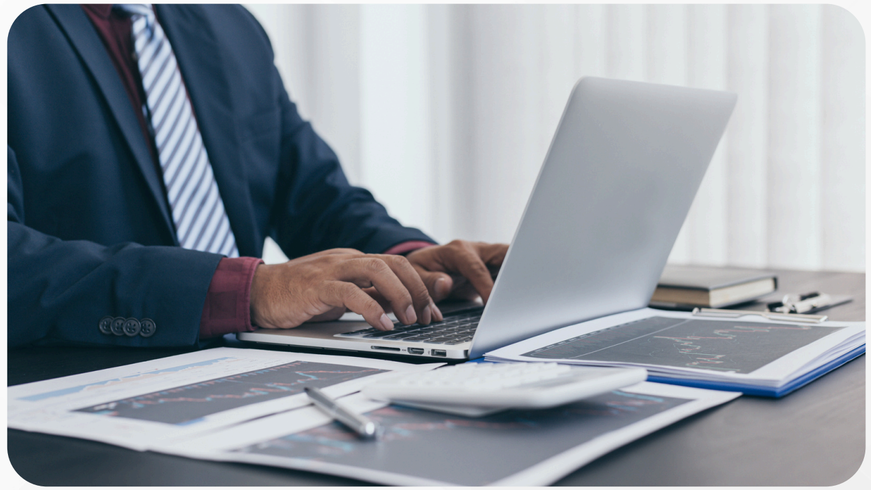
A critical section is a part of a code or resource that : must be assessed by only one node at a time.

Examples:

Updating a shared database

Writing to shared memory

Accessing a common IoT controller



COMPONENTS OF THE ALGORITHM

Each node maintains:

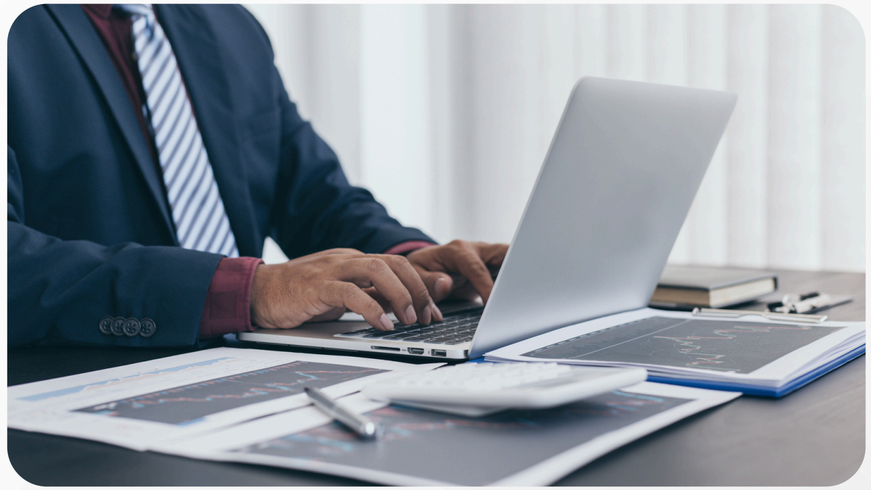
RN[i] (Request Number): Stores the latest request number from node i.

Token (Only one in the system).

LN[i]: Last request served for node i.

Queue: List of nodes waiting for the token.

*No central coordinator.
* Fully distributed.



ALGORITHM

Requesting the critical section:

(a) If requesting site S_i does not have the token, then it increments its sequence number, $RN_i[i]$, and sends a REQUEST(i, sn) message to all other sites. ("sn" is the updated value of $RN_i[i]$.)

(b) When a site S_j receives this message, it sets $RN_j[i]$ to $max(RN_j[i], sn)$. If S_j has the idle token, then it sends the token to S_i if $RN_j[i] = LN[i] + 1$.

Executing the critical section:

(c) Site S_i executes the CS after it has received the token.

Releasing the critical section: Having finished the execution of the CS, site S_i takes the following actions:

(d) It sets $LN[i]$ element of the token array equal to $RN_i[i]$.

(e) For every site S_j whose i.d. is not in the token queue, it appends its i.d. to the token queue if $RN_i[j] = LN[j] + 1$.

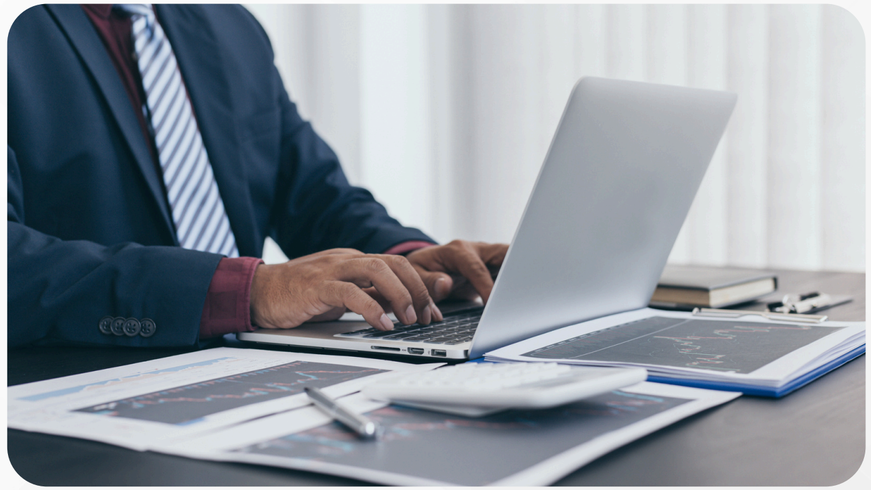
(f) If the token queue is nonempty after the above update, S_i deletes the top site i.d. from the token queue and sends the token to the site indicated by the i.d.

HOW IT IS USED IOT SYSTEMS

In IoT systems:

Multiple devices share common resources.

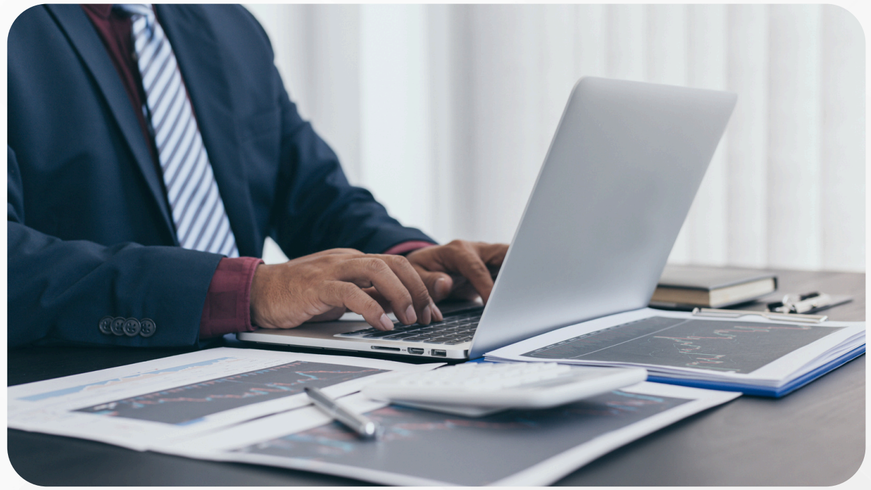
Example resources:
Cloud database
IoT gateway
Shared actuator



HOW IT IS USED IOT SYSTEMS

Why Suzuki–Kasami fits IoT:

- No single point of failure.
- Low message overhead.
- Suitable for distributed smart devices.



THANK YOU