

3. Quiz

- Sistem distribuit vs centralizat
- Control centralizat
- Control descentralizat
- Control distribuit

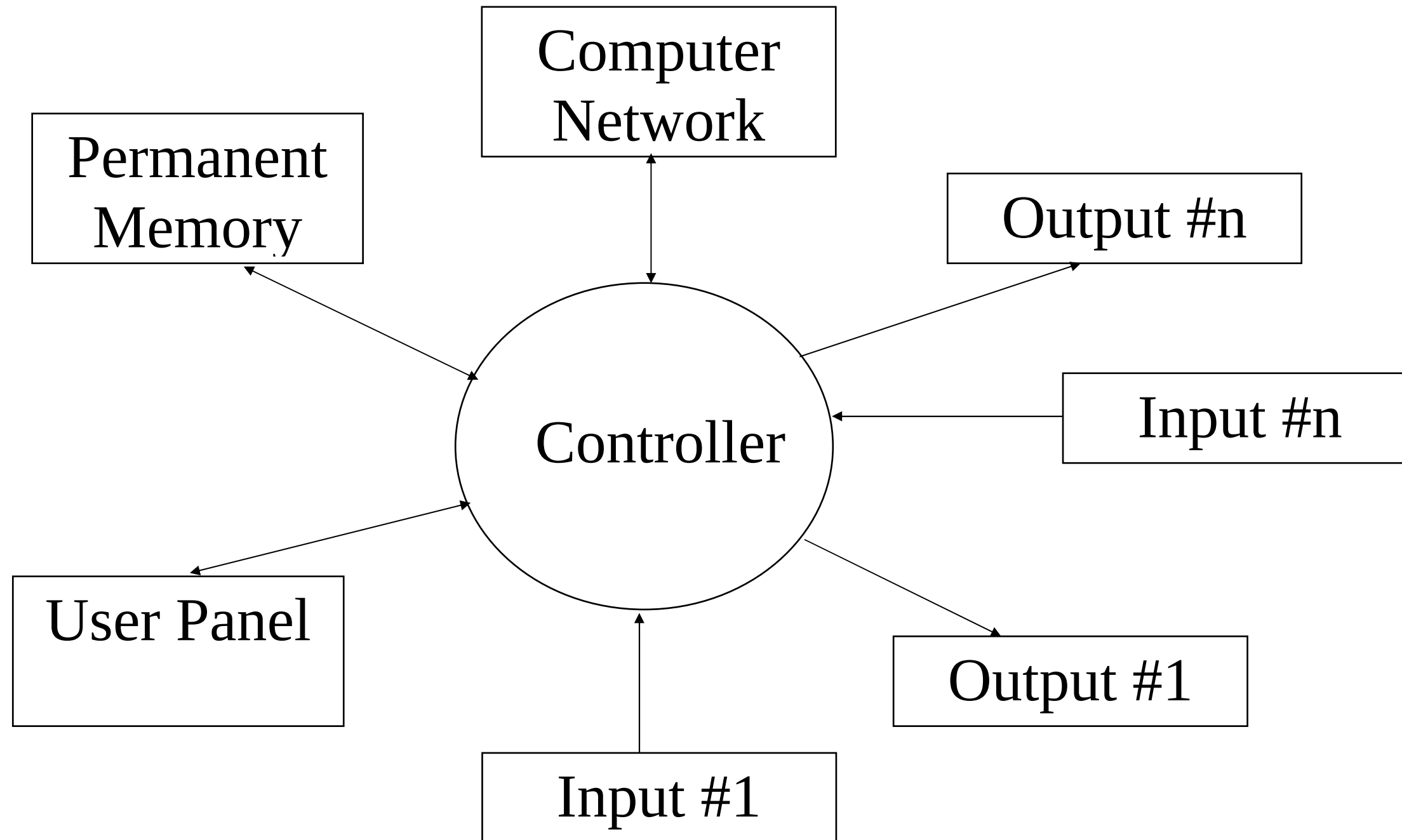
De ce distribuit ?

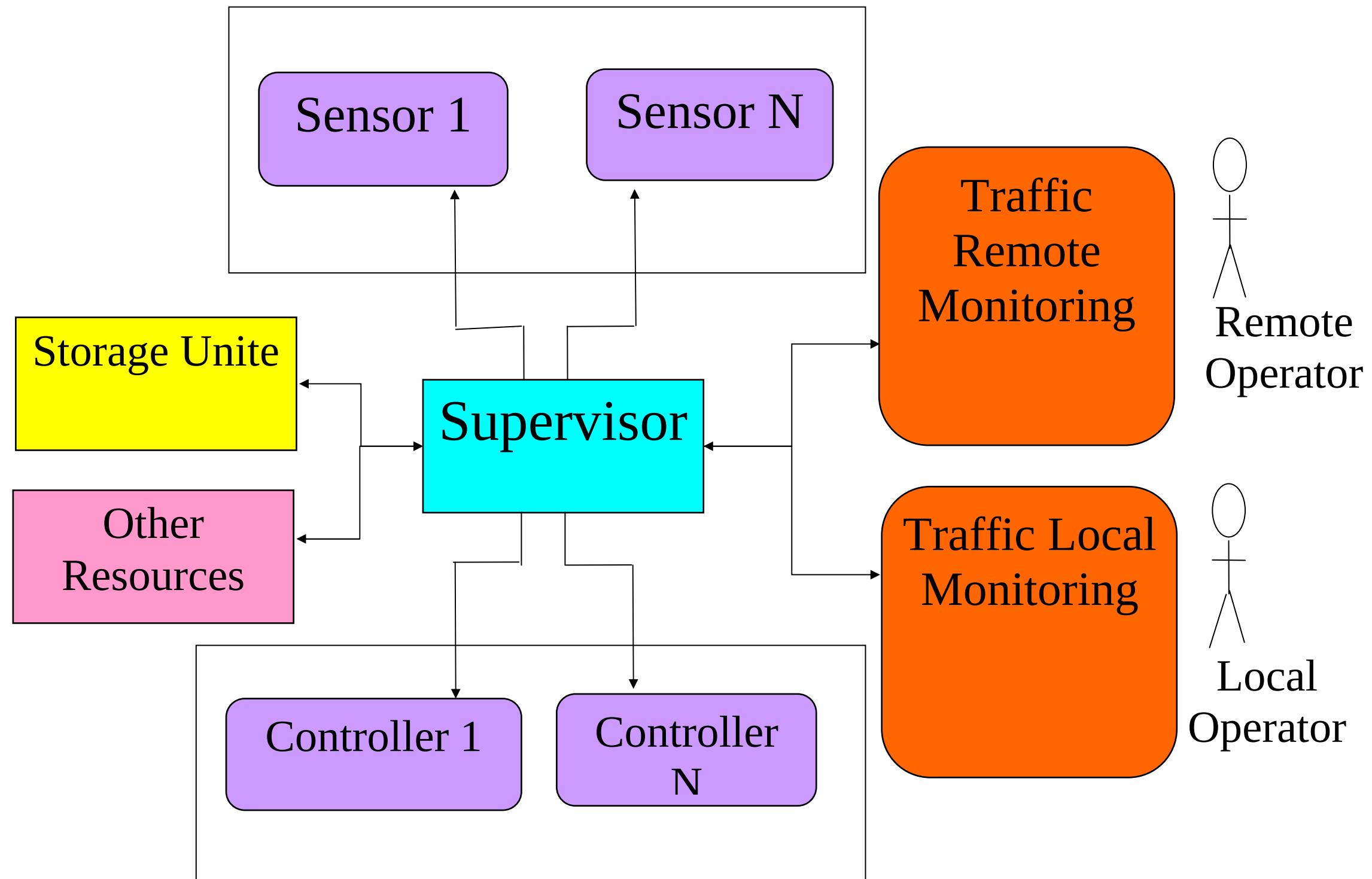
3. Sisteme de control distribuit

Arhitecturi

- 3.1. Diagrama de context
- 3.2. Descompunerea (partiționarea) task-urilor de control
- 3.3. Criterii pentru construirea arhitecturilor
- 3.4. Descompunerea planificării activităților și a utilizării resurselor
- 3.5. Arhitecturi de control distribuit cooperativ
- 3.6. Evaluarea planificării
- 3.7. Exemple de arhitecturi distribuite

3.1. Diagrame de context pentru un sistem de control





3.2. Descompunerea task-urilor de control (partiționare)

- control multistrat sau descompunere multistrat

Motive:

- *Cerințe de reactivitate diferită*
- *Distanța (local + la distanță)*
- *Capacitatea de memorie*
- *Condiții de mediu*

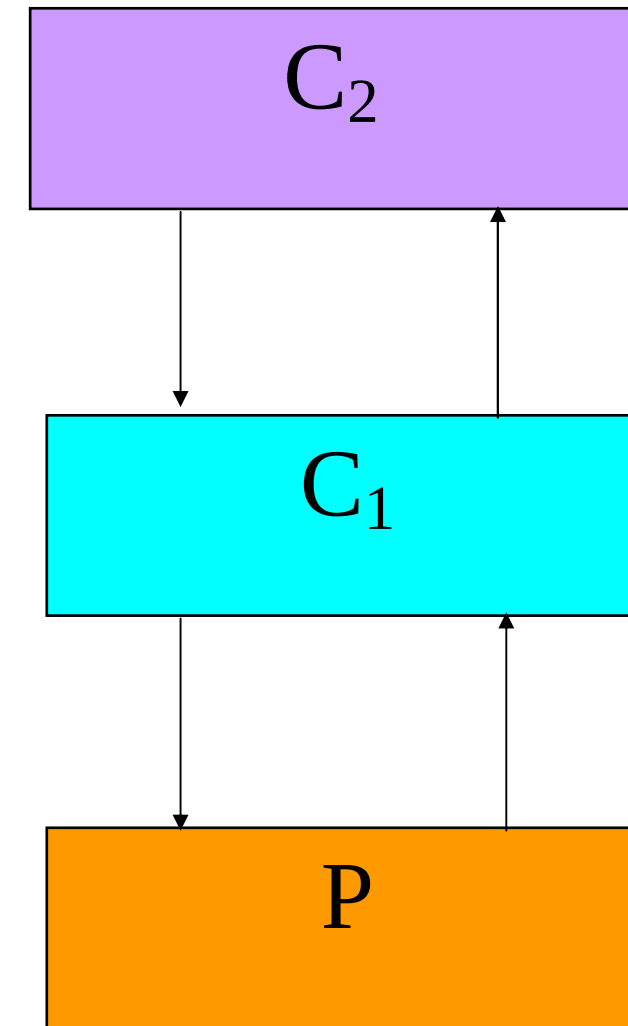


Fig. 3.3. Decompunere verticală

Motive:

- *Structura sistemului (distribuit)*
- *Distanța (local + la distanță)*
- *Capacitatea de memorie*
- *Condiții de mediu*

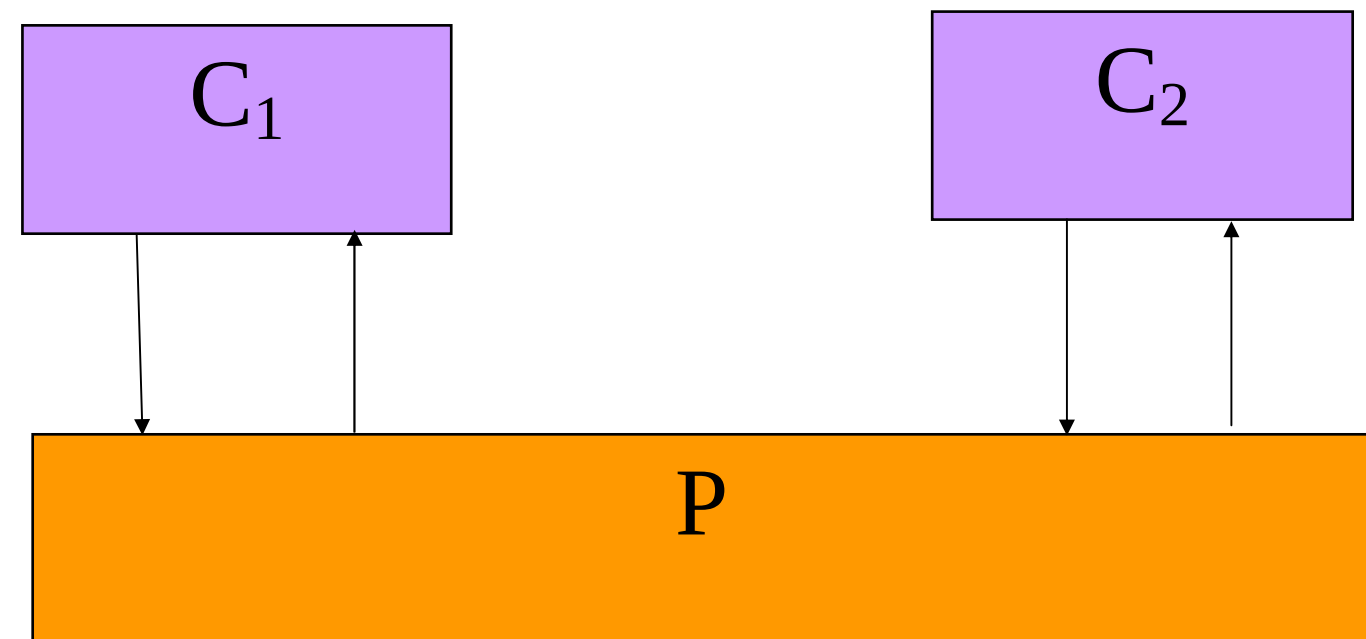


Fig. 3.4. Descompunere orizontală
- control independent
- control heterarhic (componente care comunică)

Motive:

- *Cerințe de reactivitate diferită*
- *Distanța (local + la distanță)*
- *Capacitatea de memorie*
- *Impărțirea capacității computaționale*
- *Condiții de mediu*

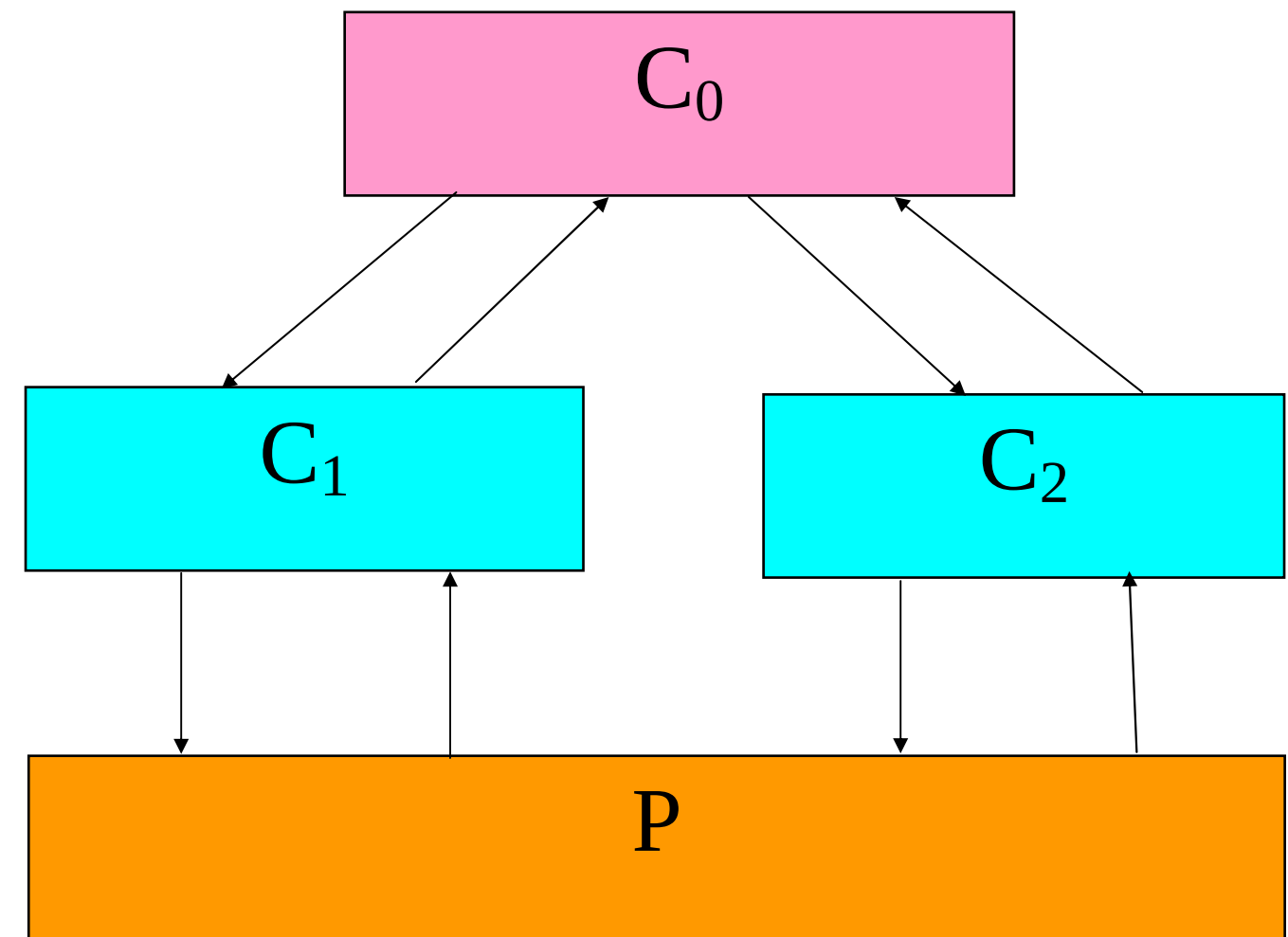


Fig. 3.5. Structură de control cu element de coordonare (ierarhic)

3.3. Criterii pentru construirea arhitecturilor

- modelare
- descompunerea sistemului
- flexibilitatea arhitecturii

Importanța modelării sistemului

Importanța descompunerii sistemului

UML are mecanisme pentru acest scop

Importanța flexibilității

- accesul controlului
- dezvoltarea bazată pe componente
- conexiune și coeziune

Metoda: Procesarea rapidă orientată pe obiecte pentru sisteme embedded

UML-RT

GIOTTO

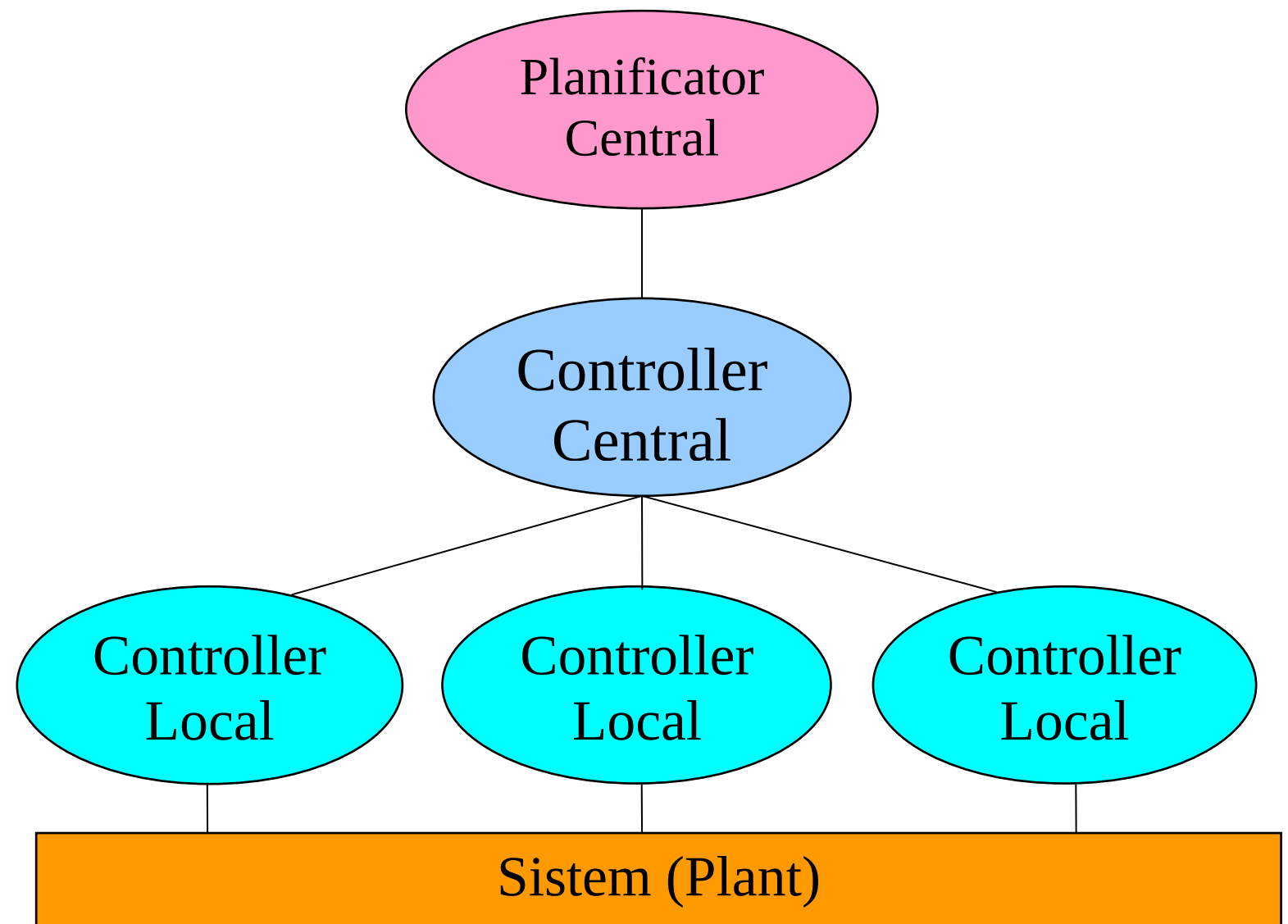
MARTE

3.4. Descompunerea planificării activităților și a utilizării resurselor

Arhitectură centralizată

Motive

- *Specificațiile sistemului
(activități planificate)
(ex. Trafic feroviar; FMS etc)*
- *Utilizarea resurselor în comun*



Arhitectura independentă și heterarhică

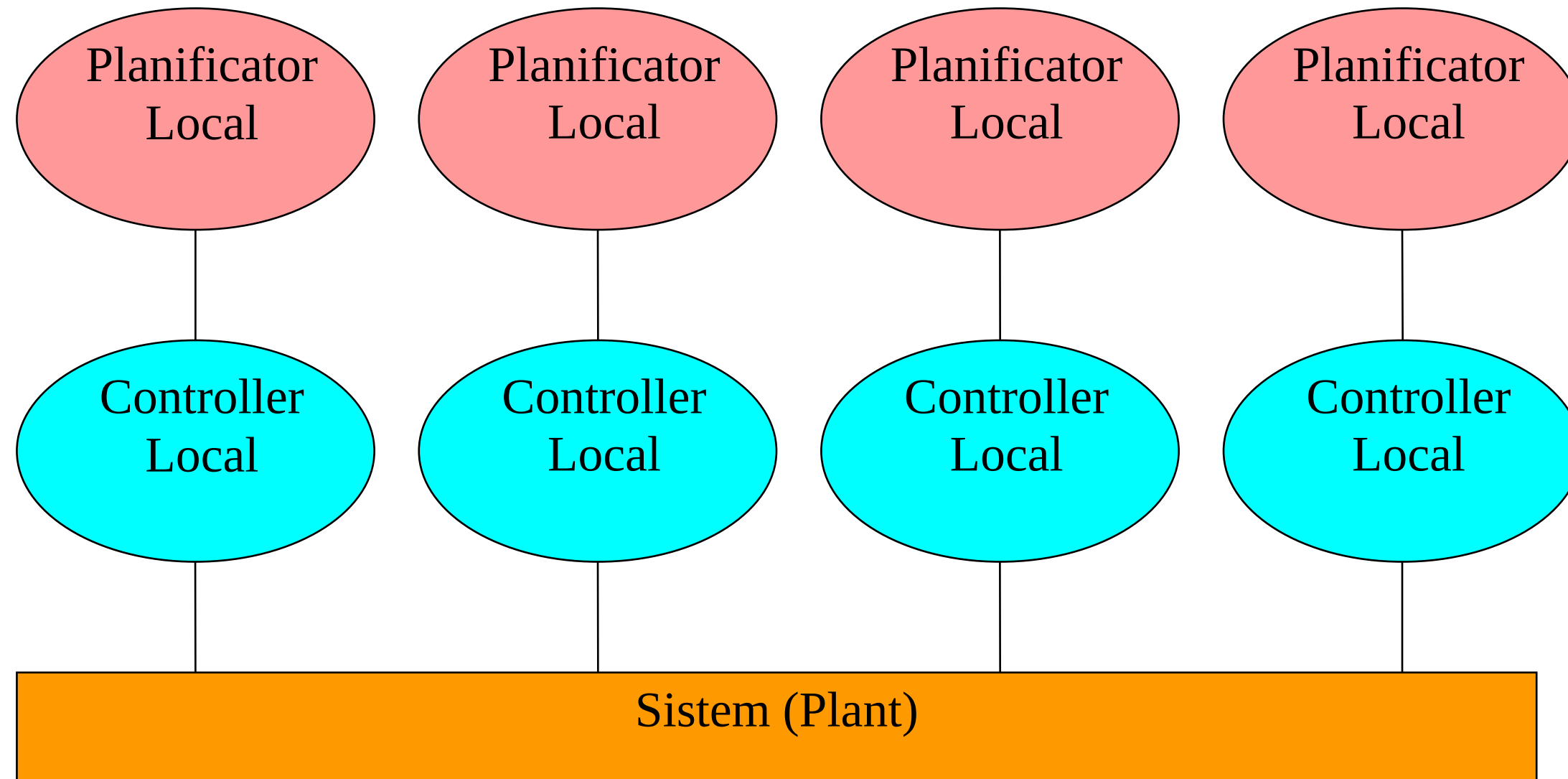


Fig. 3.7. Arhitectură heterarhică de control.

Motivele ?

3.5. Arhitecturi de sistem de control cooperativ

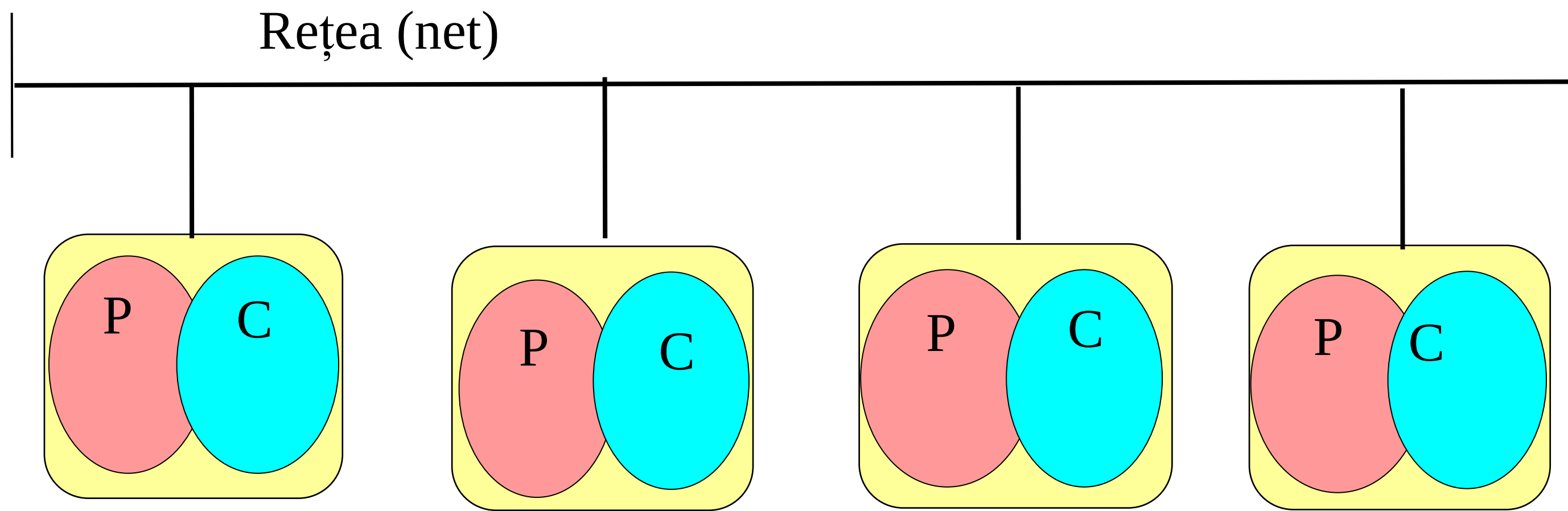


Fig. 3.8. Arhitectură cuplată puternic.

S = Planificator

C = Controler

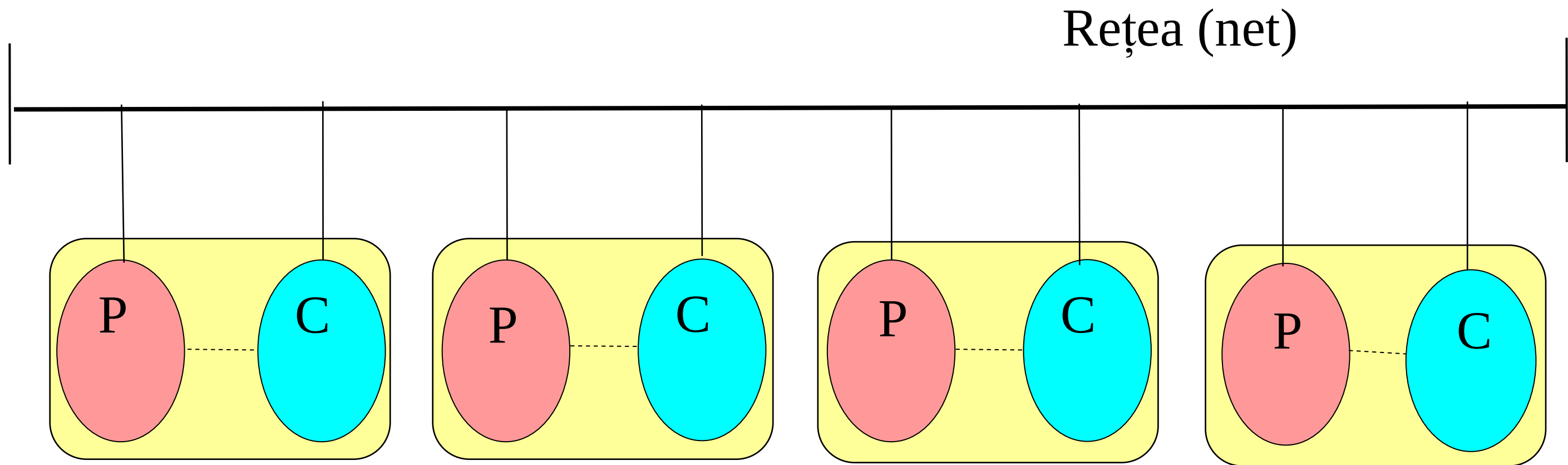
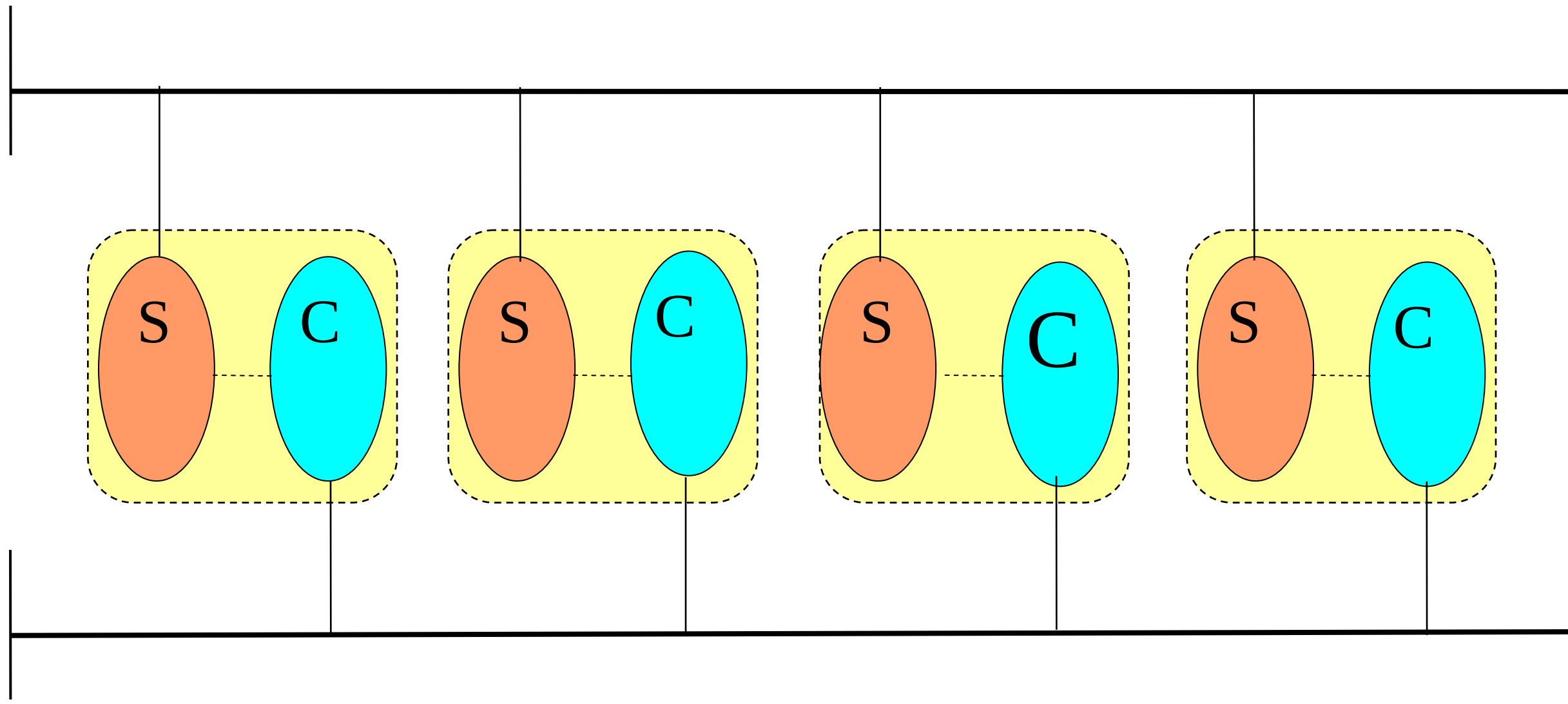


Fig. 3.9. Arhitectură cuplată slab – **Aceeași rețea.**

S = Planificator

C = Controler

Rețeaua planificatoarelor



Rețeaua Controlerelor

Fig. 3.10. Arhitectură cuplată slab – **rețele separate**

3.6. Evaluarea planificării

1) Tehnici formale

2) Tehnici bazate pe simulare:

- simulare bazată pe timp
- simulare bazată pe evenimente

Simulatoare bazate pe timp (caracterizare):

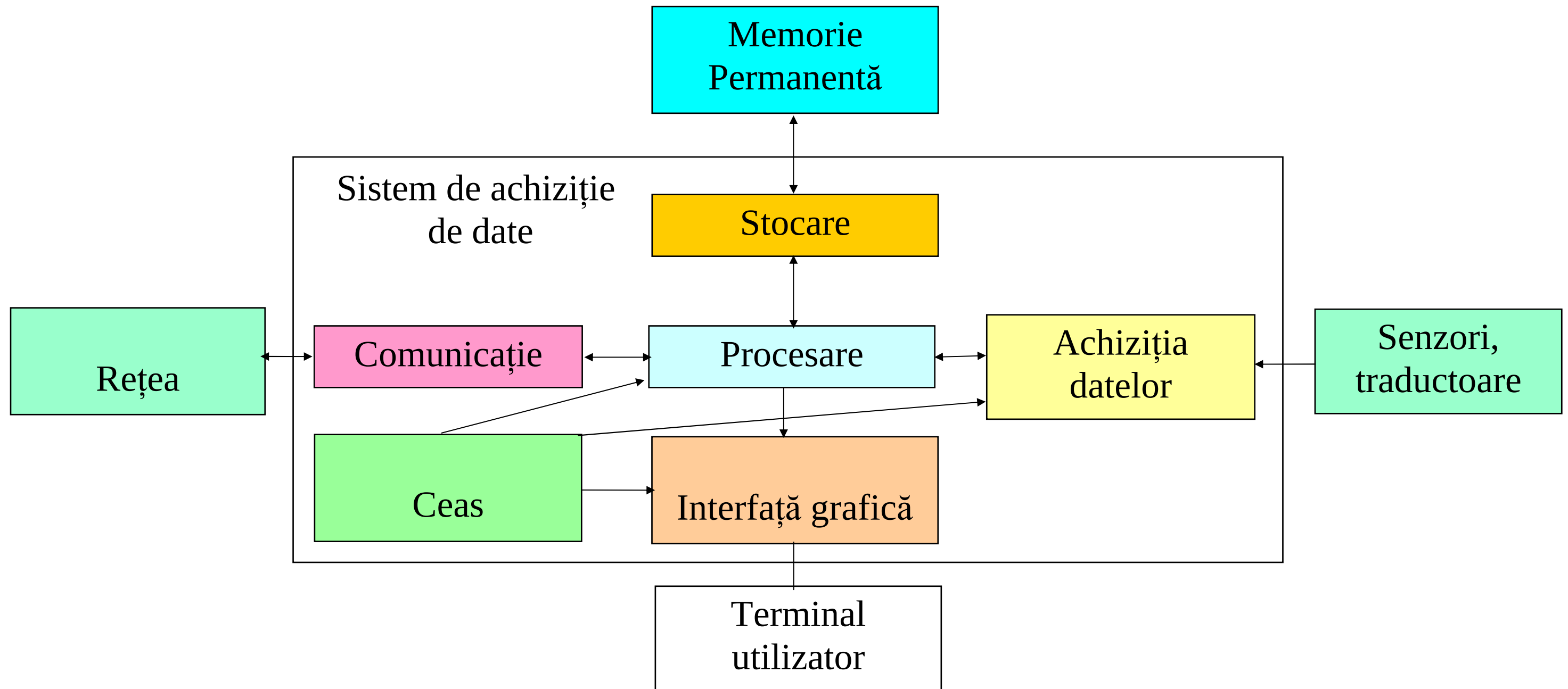
- Au mecanisme de sincronizare pentru a trimite și a primi mesaje.
- Implică efort minim pentru dezvoltare și implementare.
- Capacitatea de calcul și comunicație nu sunt folosite la maxim. Există perioade cu evenimente puține

Simulatoare bazate pe evenimente (caracterizare):

- Mecanismele de simulare sunt mai complexe (trebuie să mențină ordinea de apariție a evenimentelor în ciuda naturii distribuite a sistemului)
- Implică efort considerabil pentru dezvoltare și implementare.
- Simularea este mai rapidă decât cea bazată pe timp. Viteza de simulare este dată doar de capacitatea de calcul și cea de comunicație.

Acuratețea simulării **Controler virtual**

3.7. Exemple de arhitecturi distribuite



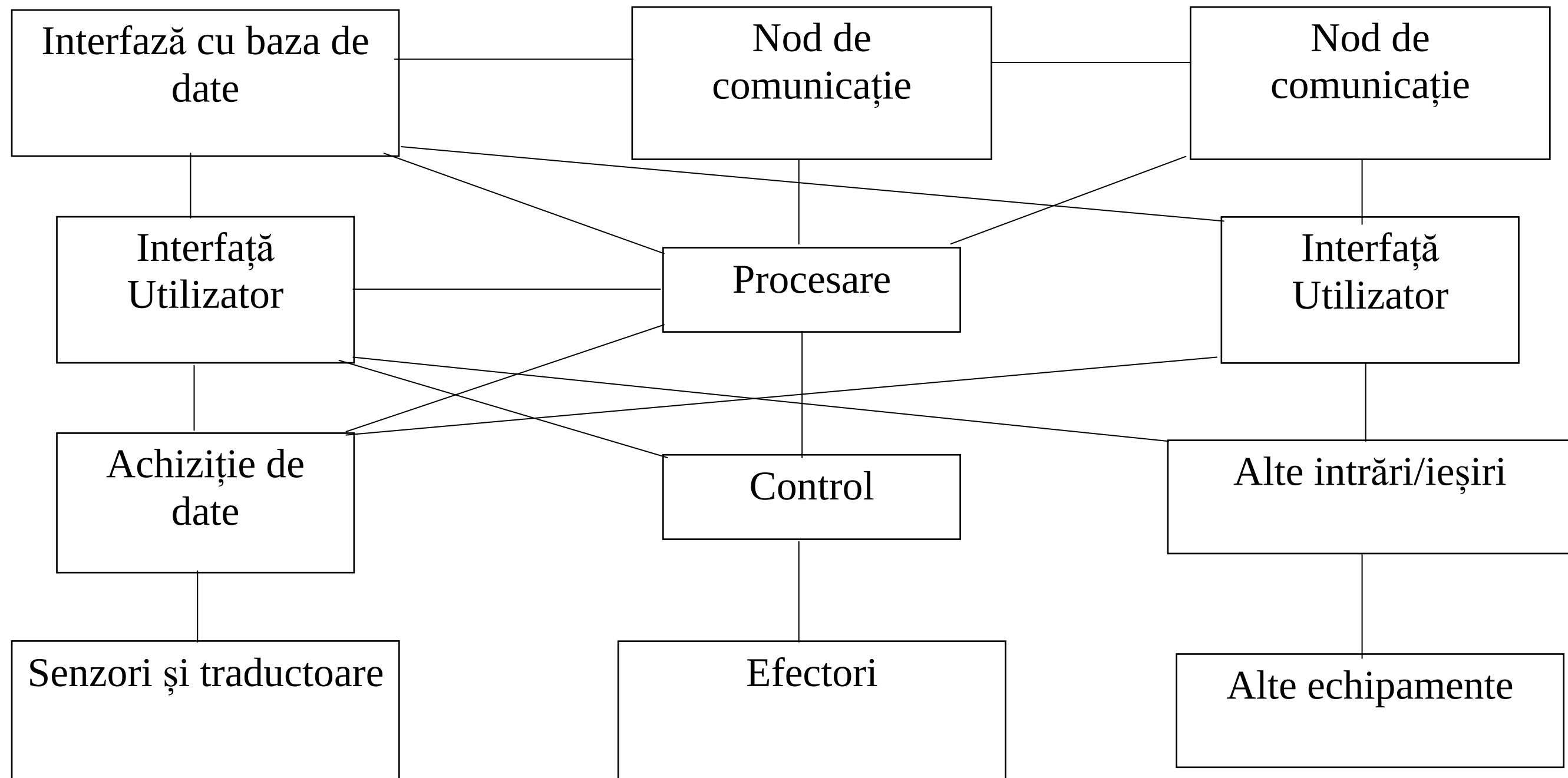


Fig. 3.11 Arhitectura generică a SCD - exemplu

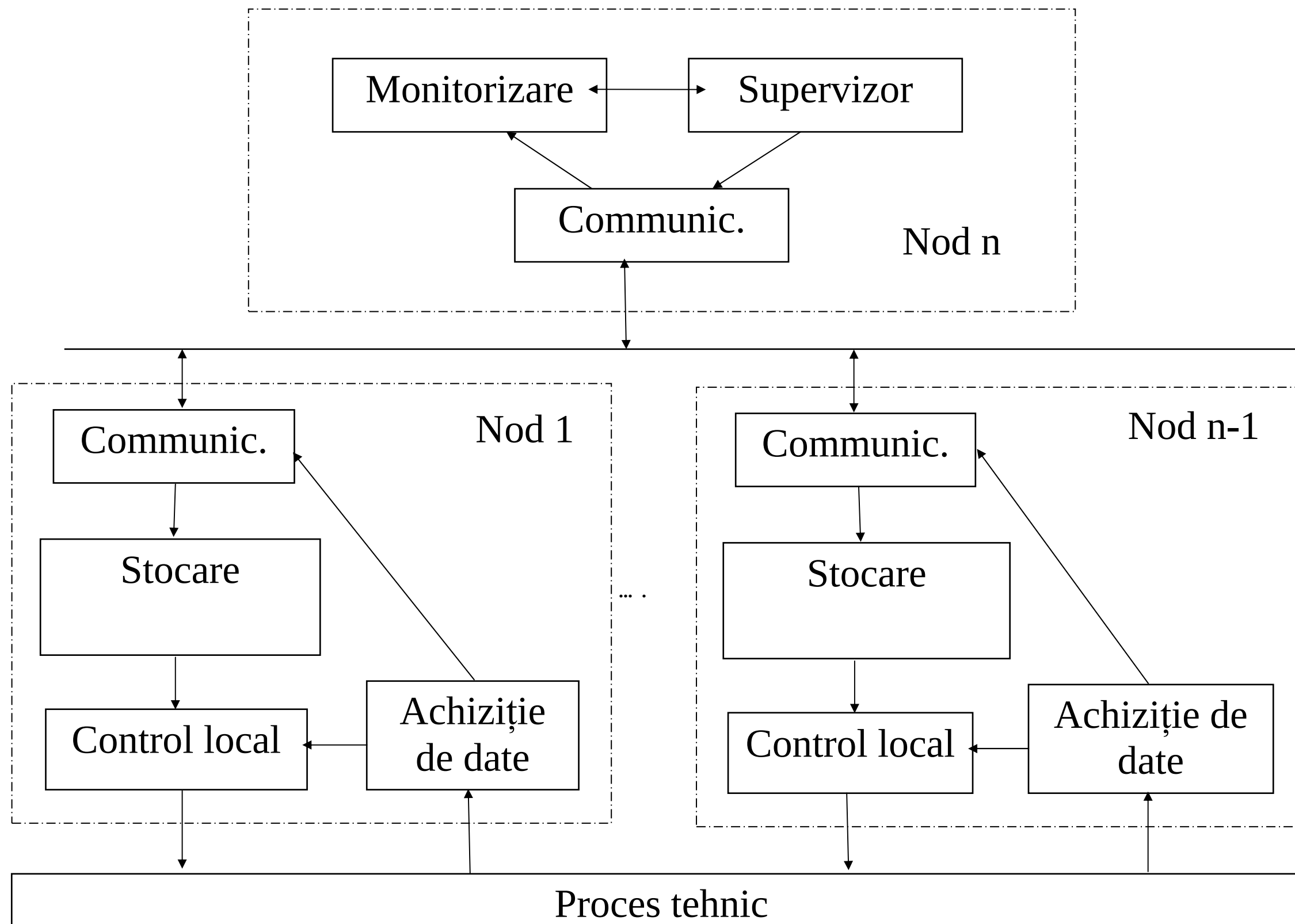


Fig. 3.12 Arhitectura generică a SCD - exemplu

Arhitectura unui sistem de control pentru traficul vehiculelor

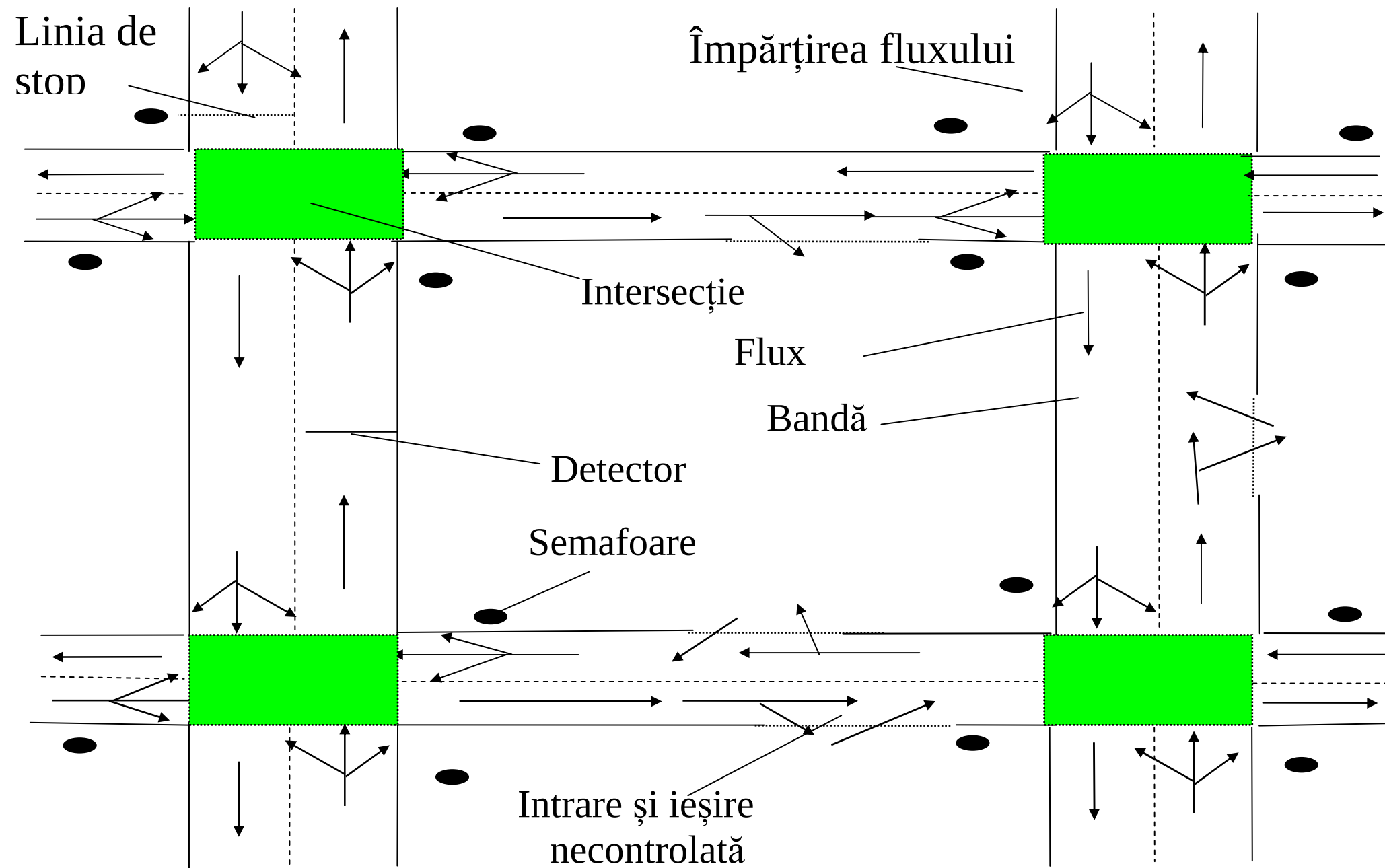


Fig. 3.14. Sistem de 4 intersecții

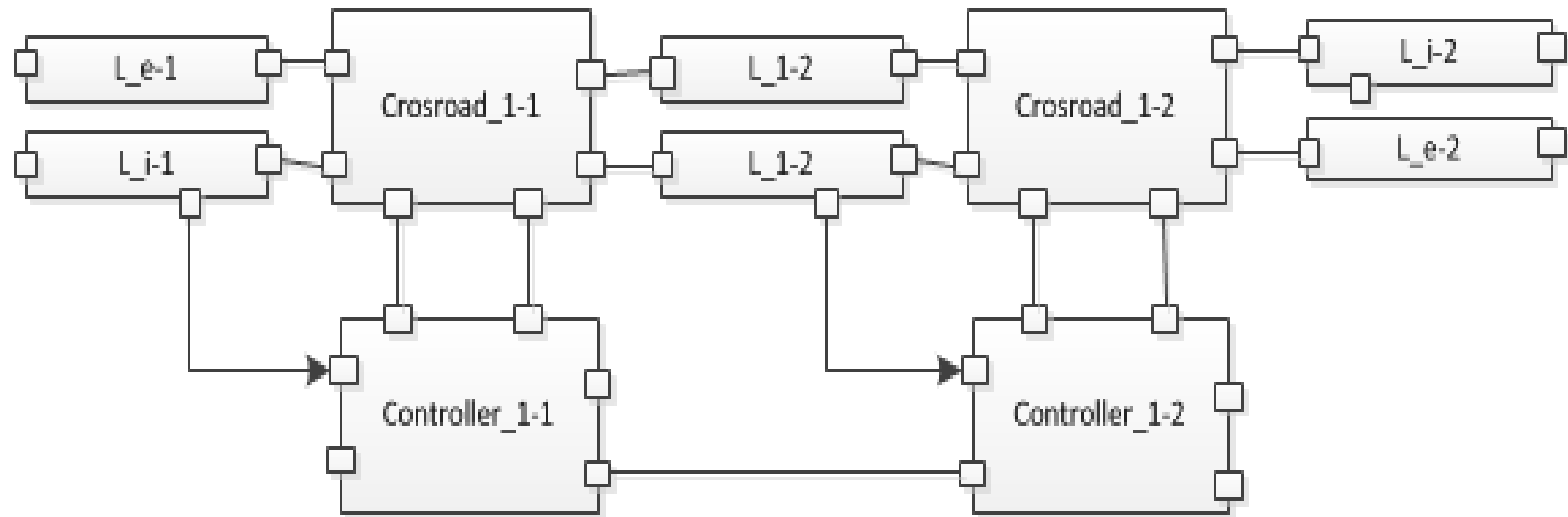


Fig. 3.15. Arhitectura sistemului de control pentru traficul rutier. Diagrama componentelor.

Arată interconexiunile între componente.

*

Sfârșit

*