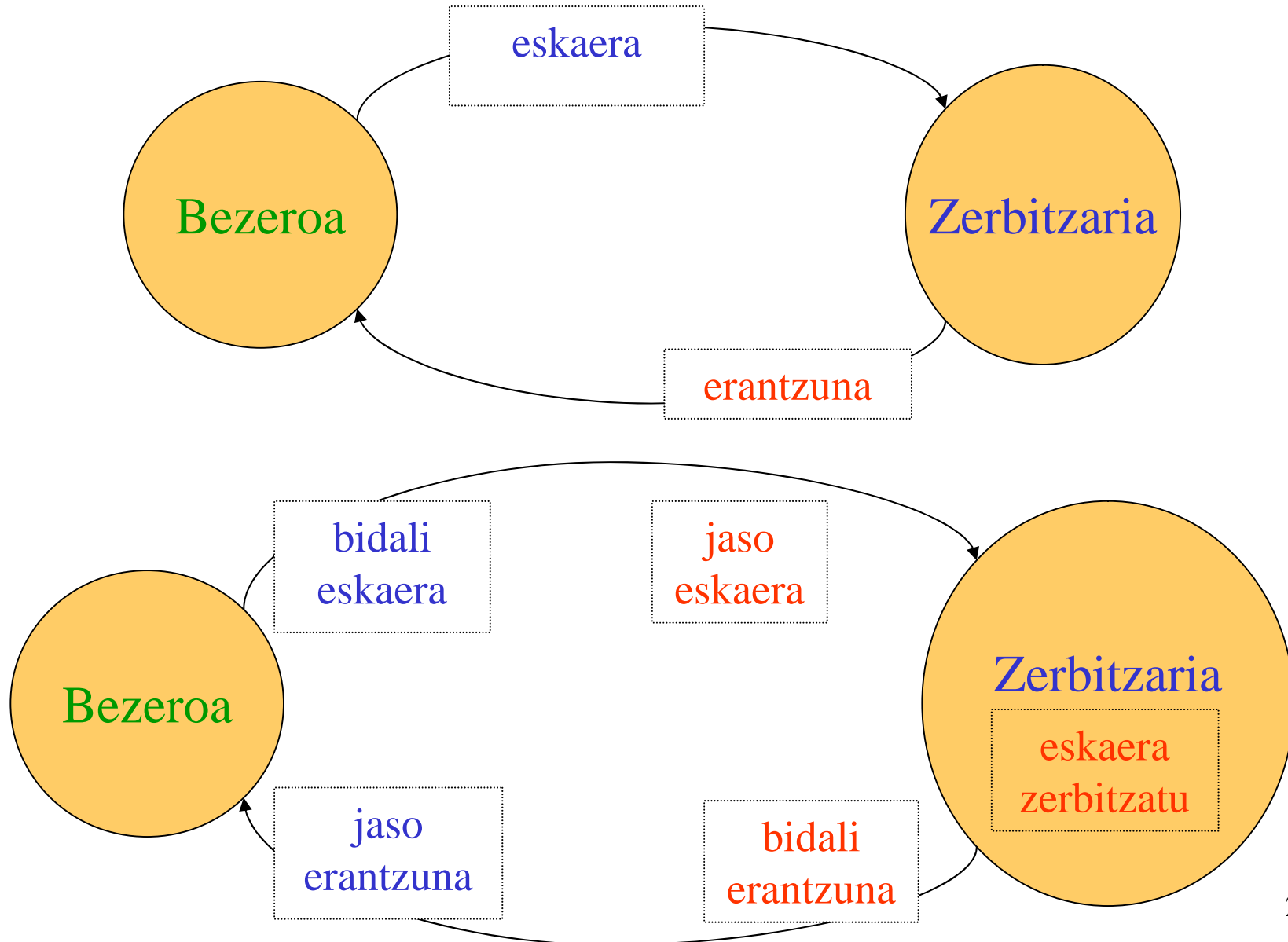


8. Gaia. Bezero-Zerbitzari eredua

1. Deskribapena.
2. Zerbitzariak, kudeatzaileak eta deabruak.
3. Zerbitzarien adibideak.
4. Zerbitzariak eta bezeroak programatzeko ereduek.

Bezero-Zerbitzari eredua



Bezero-Zerbitzari eredua

```
bezeroaren_errutina()  
{  
    bidali_eskaera();  
    jaso_erantzuna();  
}
```

```
X_zerbitzaria()  
{  
    hasieratzea();  
    while (TRUE)  
    {  
        jaso_eskaera();  
        eskaera_zerbitzatu();  
        bidali_erantzuna();  
    }  
}
```

Bezero-Zerbitzari eredua

- Bezeroa (eskaerak):

- *Sinkronoak*

- Erantzuna itxaroten du (datuak) zerbitzaritik
 - a) Eskaera jasotzen denean
 - b) Eskaera zerbitzatzaren hasten denean
 - c) Eskaera zerbitzatu (amaitu) denean

```
bezeroaren_errutina()  
{  
    bidali_eskaera();  
    jaso_erantzuna();  
}
```

- *Asinkronoak*

- Ez du erantzunik itxaroten

```
bezeroaren_errutina()  
{  
    bidali_eskaera();  
}
```

- Zerbitzaria

- *Elkarreragilea*

- Zerbitzariak aldiko eskaera bat zerbitzatzaren du

- *Konkurrentea*

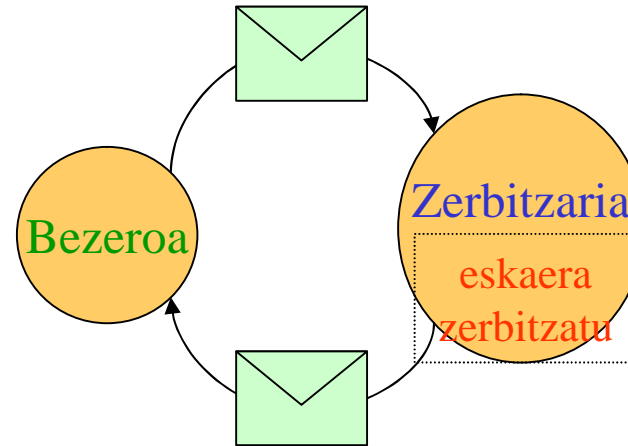
- Zerbitzariak aldi berean eskaera bat baino gehiago zerbitzatu ditzake (eskaera bakoitza beste prozesu bati pasatuz)

Bezere-Zerbitzari ereduak

- Zerbitzaria

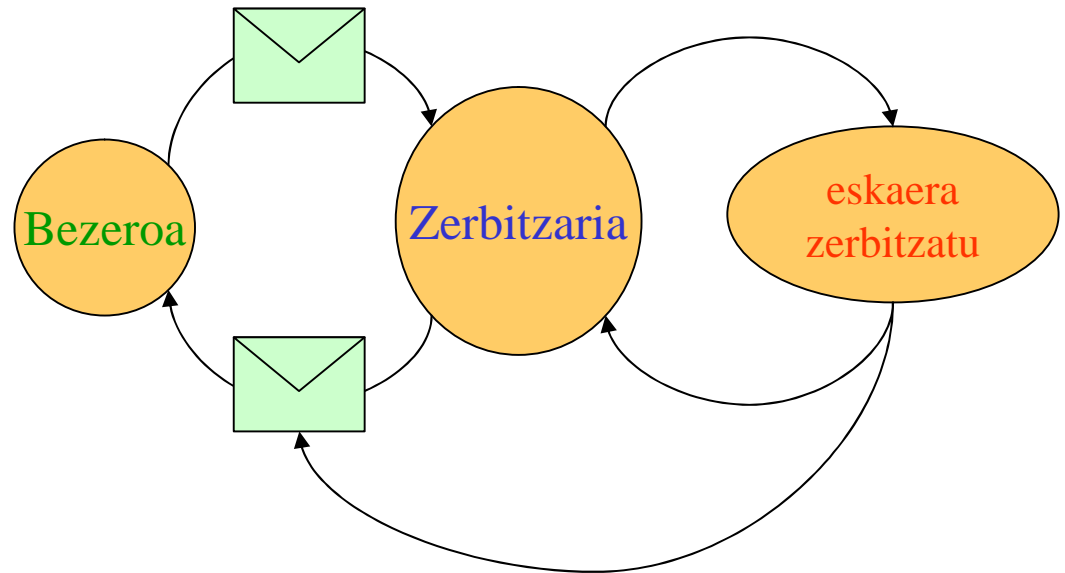
- *Elkarreragilea*

- Zerbitzaria berak burutzen du eskaera



- *Konkurrentea*

- Zerbitzariak beste prozesu bati (umea ala ez) agintzen dio eskaera burutzea



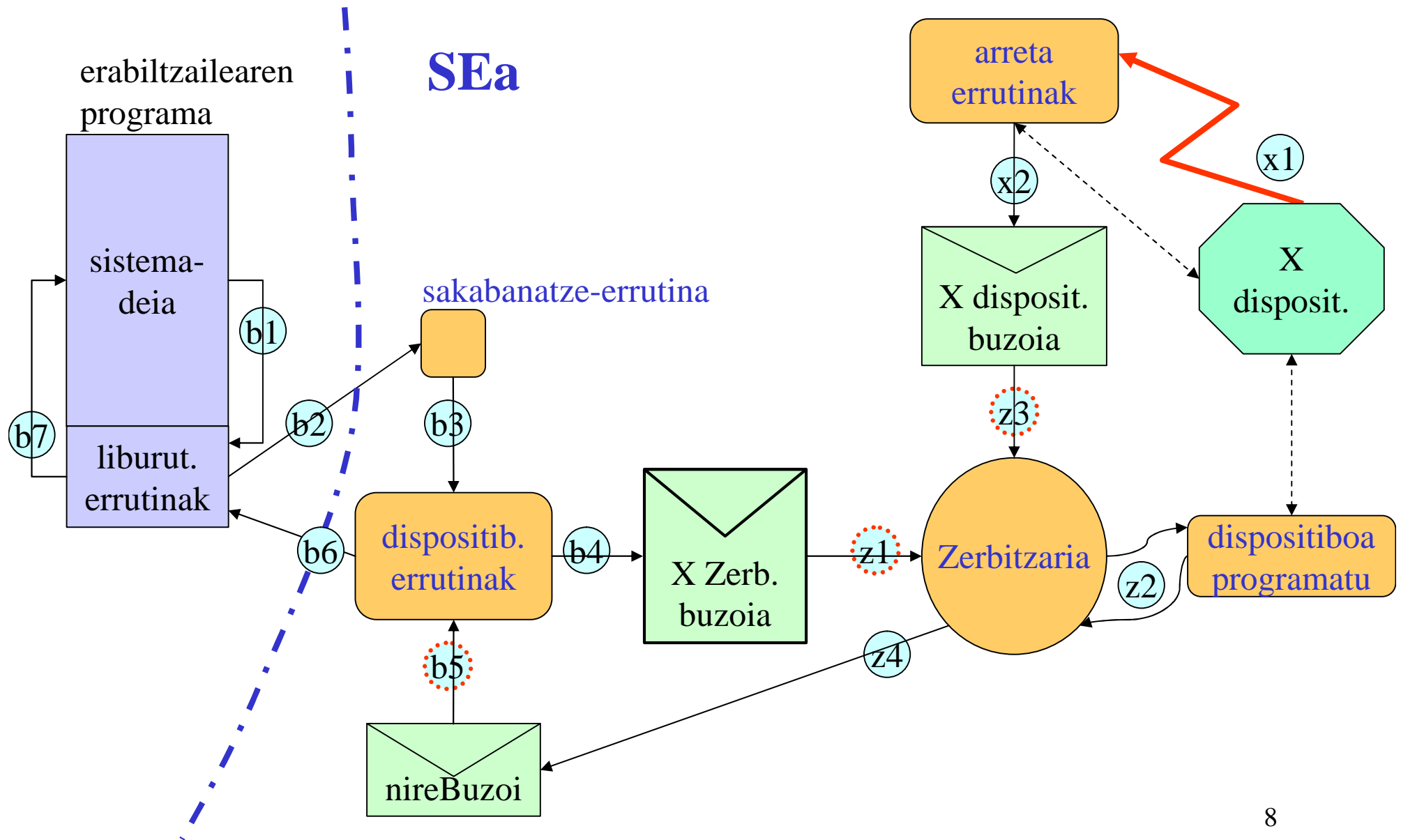
Bezeroaren errutinaren adibidea (sinkronoa/asinkronoa)

```
bezeroaren_errutina(int asin_sin, int buzoi_espl)
{
    if (asin_sin == SINKRO) { /* SINKRONOA */
        eraiki_eskaera(param, nire_buzoia, &eskaera);
        bidali_eskaera bidali(X_Zerbitzari_buzoia, &eskaera);
        jaso_erantzuna jaso(nire_buzoia, &erantzuna);
    } else { /* ASINKRONOA */
        eraiki_eskaera(param, buzoi_espl, &eskaera);
        bidali_eskaera bidali(X_Zerbitzari_buzoia, &eskaera);
    }
}
```

Zerbitzariaren adibidea

```
X_Zerbitzaria()  
{  
    hasieratu_egiturak(X_Zerbitzaria);  
    hasieratu_buzoiak(X_Zerbitzaria);  
    while (TRUE)  
    {  
        jaso  
        eskaera    jaso(X_Zerbitzari_buzoia, &esk);  
                  esk_param_lortu(esk, &param, &bez_buzoia);  
  
        eskaera  
        zerbitzatu zerbitzatu_eskatutako_eragiketa(param);  
  
        bidali  
        erantzuna  bidali(bez_buzoia, "ESK_AMAIERA");  
    }  
}
```

Bezero-Zerbitzari ereduaren adibidea: S/I-ko eskaeren kudeaketa



Bezero-Zerbitzari ereduaren adibidea: S/I sistema-deia (adib.: *read*)

```
X_dispositiboaren_errutina(char *bektore, int luz,  
                             int asin_sin, int buzoi_espl)  
{  
    if (asin_sin == SINKRO) {  
        eraiki_eskaera(bektore, luz, nire_buzoia, &eskaera);  
        bidali(X_Zerbitzari_buzoia, &eskaera); (b4)  
        jaso(nire_buzoia, &erantzuna); (b5)  
        return(erantzuna.itzulera_kodea); (b6)  
    } else {  
        eraiki_eskaera(bektore, luz, buzoi_espl, &eskaera);  
        bidali(X_Zerbitzari_buzoia, &eskaera); (b4)  
    }  
}
```

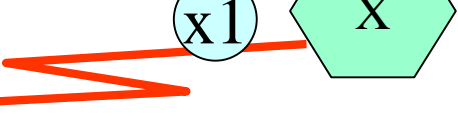
S/I-ko zerbitzaria

```
X_Zerbitzaria()
{
    struct eskaera esk, esk_emaiza;
    int luz, X_Zerbitzari_buzoi, X_disp_buzoi, bez_buzoi;
    char *bektore;
    hasieratu_egiturak(X_Zerbitzaria);
    hasieratu_buzoiak(X_Zerbitzaria);
    while (TRUE) {
        jaso(X_Zerbitzari_buzoi, &esk);           z1
        esk_param_lortu(esk, bektore, &luz, &bez_buzoi);
        X_dispositiboa_programatu(bektore, luz);  z2
        jaso(X_disp_buzoi, &esk_emaiza);         z3
        bidali(bez_buzoi, &esk_emaiza);          z4
    }
}
```



eskaera zerbitzatu

Arreta- eta sinkronizazio-errutinak

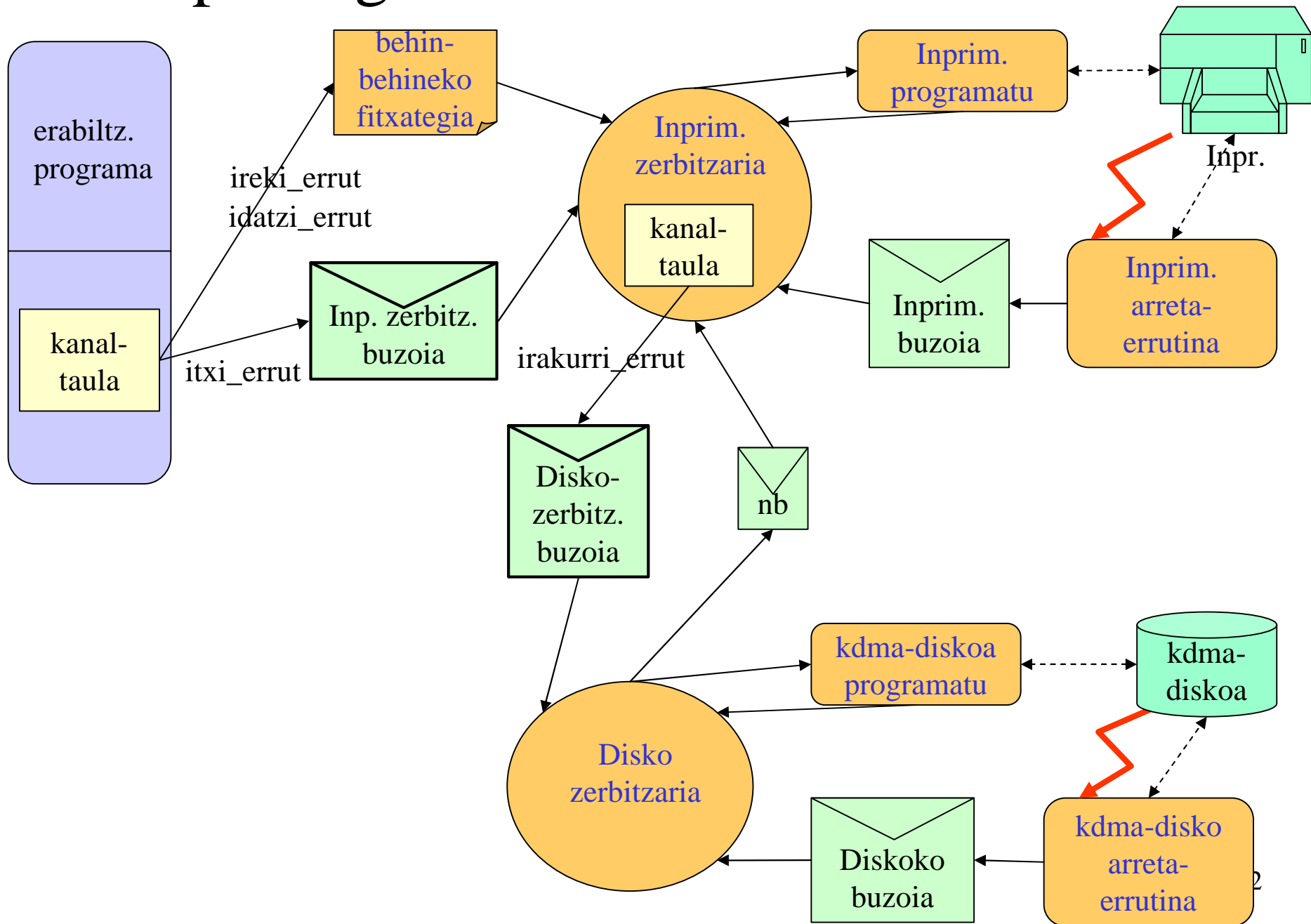
```
X_dispositiboaren_arreta_errutina() {  
    struct eskaera esk_emaitza;  
  
    irakurri_edo_idatzi_datua(); /* aztertu kontroladorea */  
    if (AMAIERA) {  
        esk_emaitza.itzulera_kodea = itzul_kodea;  
        bidali(X_disp_buzoi, &esk_emaitza);  
    }  
}
```



```
sinkronizazio_errutina(int buzoi_espl) {  
    struct eskaera esk_emaitza;  
  
    jaso(buzoi_espl, &esk_emaitza);  
    return(esk_emaitza.itzulera_kodea);  
}
```



Inprimagailuaren kudeaketaren eskema



Inprimagailuaren kudeaketa (ireki_errutina)

```
int ireki_errutina(char *izena)
{
    if (inprimagailua_da(izena)) {
        behin_behineko_fitxategia_sortu(iz);
        kanala = inprimagailua_fitxat_ireki_errutina(iz);
        return(kanala);
    } else {
        /* kanala = X_ireki_errutina(izena); X disp. arabera */
        return(kanala);
    }
}
```

Inprimagailuaren kudeaketa (itxi_errutina)

BEZEROA DA!

```
int itxi_errutina(int kanala)
{
    if (inprimagailuari_lotuta(kanala)) {
        izena_lortu(kanala, izena);
        eraiki_eskaera(izena, &eskaera);
        bidali(Inprim_Zerbitzari_buzoi, &eskaera);
    }
    kanala_askatu(kanala);
}
```

Inprim. kudeaketa - Zerbitzaria

```
Inprimagailu_Zerbitzaria()
```

```
{
```

```
...
```

```
hasieratu_buzoiak(Inprim_Zerbitzari_buzoi, Inprim_buzoi);
```

```
while (TRUE) {
```

```
    jaso(Inprim_Zerbitzari_buzoi, &eskaera);
```

```
    kanala = ireki_errut(eskaera.izena);
```

```
    while ((luz = irakurri_errutina(kanala, bekt, TAM)) != 0) {
```

```
        Inprimagailua_programatu(bekt, &luz);
```

```
        jaso(Inprim_buzoi, &esk_amaiera);
```

```
    }
```

```
    itxi_errutina(kanala);
```

```
    ezabatu_errutina(eskaera.izena);
```

```
}
```

```
}
```

eskaera
zerbitzatu

SE multiprogramatu baten funtzionamendua

