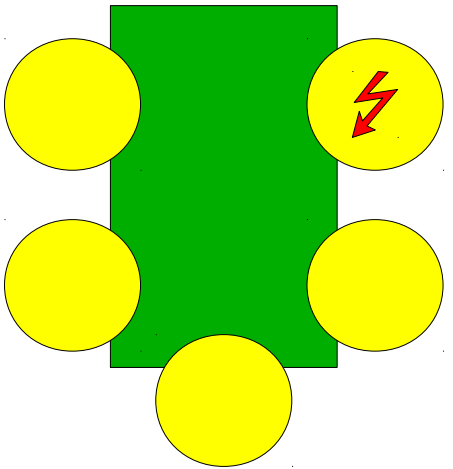
# Pluginy v C

## Jak to dělá uCollect

Michal Vaner • michal.vaner@nic.cz • 13. 11. 2015

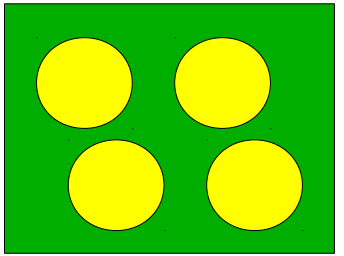
# Proč pluginy?

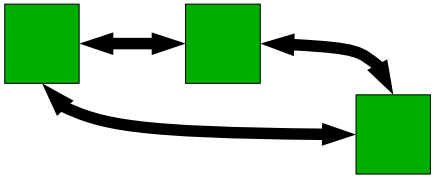
- Rozdělení kódu na samostatné celky
- Možnost přispění třetí strany
- Úpravy částí kódu za běhu
- Tolerance k chybám

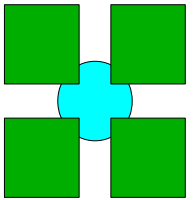


# Jde to jinak?

- Monolitická architektura
  - Jednoduché
  - Nesplňuje požadavky
- Oddělené procesy
  - Výkon
- Oddělené procesy a sdílená paměť
  - Synchronizace, uklízení, draci…





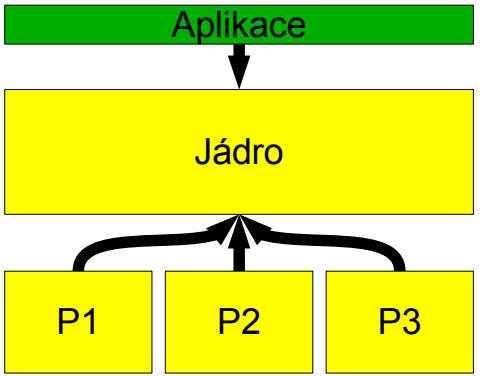


# Výzvy

- Kompilovaný jazyk
- Ruční správa paměti
- Všechny chyby jsou fatální
- Poziční ABI

# Provedení v uCollectu

- Knihovna jádra
  - Většina funkcionality
  - Rozhraní pro pluginy
  - Pomocné funkce
- Minimalistická aplikace obalující knihovnu
- Knihovny pluginů, které linkují vůči jaderné



# Načtení

- Řízené konfiguračním souborem
- Knihovna `<dlfcn.h>`
  - `dlopen` – načtení knihovny
  - `dlsym` – nalezení vstupního bodu
- Spuštění vstupního bodu
- Struktura s popisem a callbacky

# Odstranění

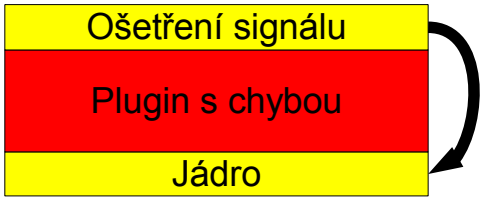
- Při změně a načtení konfigurace
- Odstranění všech callbacků (⚠)
- Odstranění všech zdrojů
- `dlclose`

# Update

- Jen odstranění starého a načtení nového
- Zatím bez zachování dat pluginu
- Ostatní pluginy nejsou ovlivněny

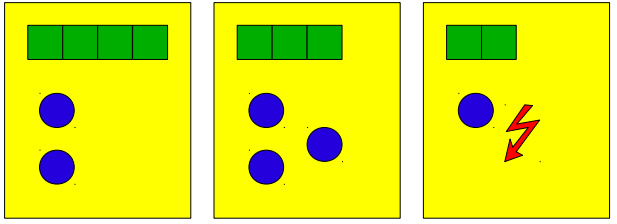
# Řešení chyb

- Chyby oznamovány signály
- Odchycení, `longjmp` ven z pluginu
- Odpovídající plugin ukončen/restartován
  - Limit na počet restartů
  - Automatické vrácení zdrojů



# Správa zdrojů

- Alokace z memory poolů
  - Memory pool patří vždy některému pluginu
  - Možnost zrušit celý pool
- Registrace file deskriptorů
- Registrace potomků



# Děkuji za pozornost

**Michal Vaner • michal.vaner@nic.cz**