Przykłady DFS z lotu ptaka :)

- NFS
- AFS
- Coda
- GoogleFS
- ZFS

NFS

- Network File System – sieciowy system plików
- Stworzony przez Sun Microsystems
- Dostępny dla, m.in.: Unix, Windows, OS/2, Mac OS
- Pracuje w środowisku heterogenicznym
- Obsługa stacji bezdyskowych

XDR – eXternal Data Representation

- Jednolity format reprezentacji liczb całkowitych: 32-bitowe big-endian
- Kodowanie liczb rzeczywistych w formacie IEEE (float i double)

Architektura NFS

Koncepcja
- Współdzielenie plików pomiędzy dowolną liczbą komputerów
- Dedykowany dla sieci lokalnych (nowsze implementacje wspierają sieci rozległe)
- Każdy komputer może być jednocześnie klientem i serwerem

Eksportowanie danych
- Udostępnianie katalogów ze wszystkimi podkatalogami

Importowanie danych
- Montowanie zdalnych katalogów przez klientów
- Zamontowany katalog staje częścią lokalnej hierarchii katalogów serwera
- Nierozróżnialność plików lokalnych i zdalnych

Cechy NFS

- Przezroczystość dostępu
- Przezroczystość położenia — poprzez spójną konfigurację montowania zdalnych katalogów
- Przezroczystość awarii
- Częściowa przezroczystość wydajności — intensywne wykorzystanie pamięci podręcznych
- Przezroczystość wędrówki

NFS nie oferuje

- Przezroczystości zwielokratniania
- Przezroczystości współbieżności
- Nieograniczonej skalowalności

Protokół NFS

Zbiór zdalnych procedur (RPC)

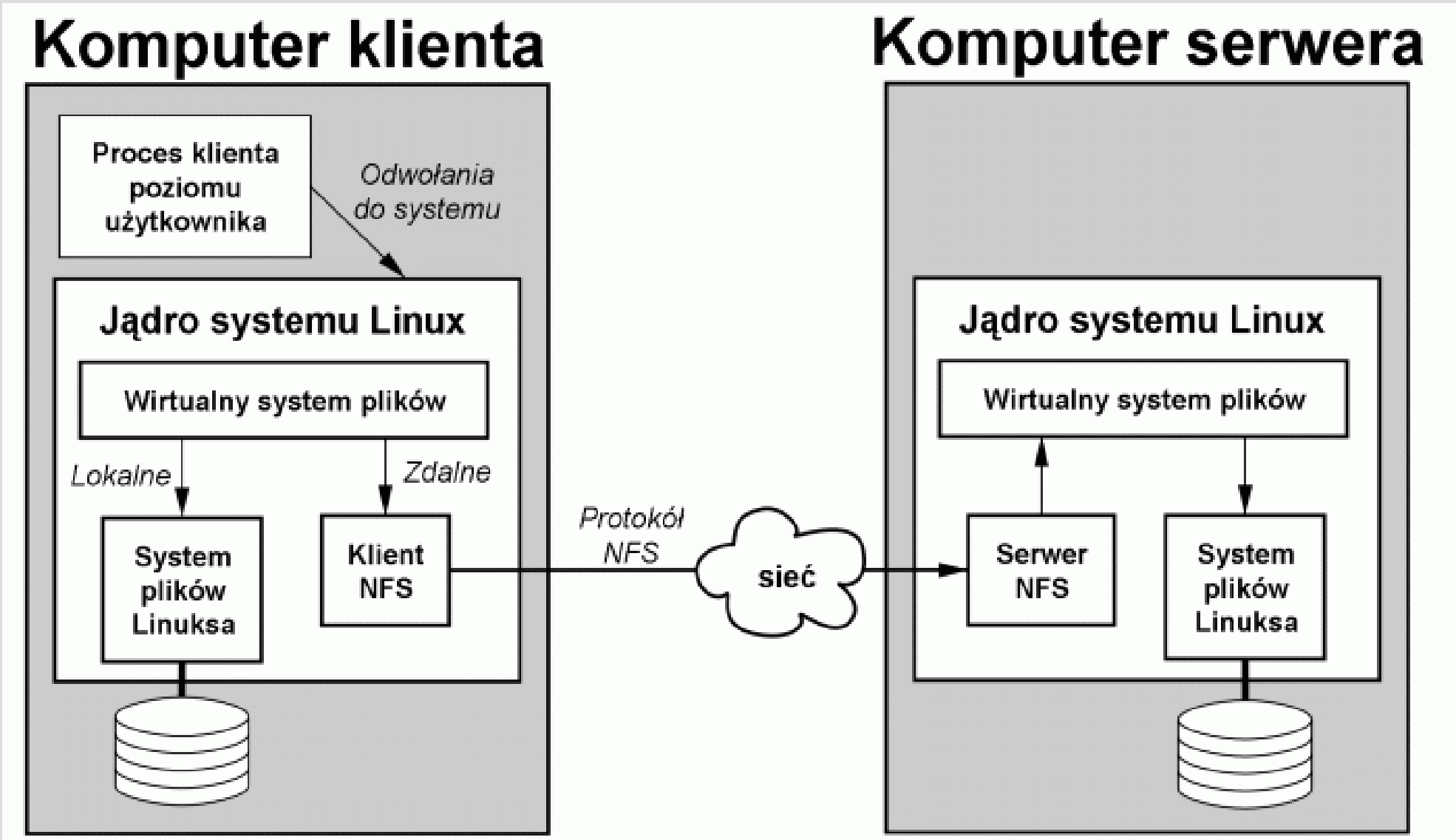
- Klient wysyła żądanie
- Serwer sprawdza uprawnienia i próbuje wykonać zadanie
- Serwer odsyła rezultaty lub informacje o błędach

Implementacja NFS (1)

Warstwy implementacji NFS

- Odwołania systemowe (open, read, write, close)
- Wirtualny system plików, VFS – ang. Virtual file system
- Właściwy klient/serwer NFS

Implementacja NFS (2)



Implementacja NFS (3)

- Transfer danych
  - Zawsze w paczkach po 8KB
  - Czytanie z wyprzedzeniem przez klienta
  - Opóźniony zapis
- Spójność pamięci podręcznych klientów
  - Kontrola znaczników czasowych przy każdym dostępie
  - Pamięć danych: ważna maksymalnie 3s
  - Pamięć atrybutów: ważna maksymalnie 30s
- Problem spójności nie jest do końca rozwiązany

NFS v4

- Domyślnie TCP, pojedynczy port, jedno połączenie
- Zintegrowane protokoły montowania i dostępu do plików
- Zoptymalizowany transfer danych
- Bezpieczeństwo
  - Kodowanie RPC
- Replikacja

AFS – Andrew File System

- Przezroczysty dostęp do zdalnych, współdzielonych plików
- Skalowalność
- Przechowywanie całych plików w trwałych pamięciach podręcznych

Scenariusz działania

- Wywołanie funkcji open, skopiowanie pliku z serwera, wykonanie lokalnie funkcji open
- Dostęp do pliku realizowany całkowicie lokalnie (read,write)
- Zamknięcie pliku.
- Zmodyfikowana kopia jest wysylana na serwer

Wydajność AFS

- **Charakterystyka plików**:
  - Większość plików jest mała (<15KB)
  - Odczyty są znacznie częstsze od zapisów
  - Większość plików jest modyfikowana przez jednego użytkownika
  - Odwołania do plików są skumulowane
- **Rozwiązanie efektywne dla większości plików**:
  - Tylko do odczytu
  - Rzadko modyfikowanych
  - Nie współdzielonych
- Duże, modyfikowane i współdzielone => bazy danych

Architektura AFS

Implementacja funkcji open

- Venus: sprawdzenie obecności pliku w pamięci podręcznej. Jeśli nie ma – wysłanie zamówienia do procesu Vice
- Przesłanie kopii pliku do klienta. Rejestracja pobrania kopii na serwerze
- Lokalne otwarcie pliku i zwrócenie deskryptora

Implementacja funkcji close

- Zamknięcie pliku z lokalną kopią
- Przesłanie zmodyfikowanej wersji do serwera
- Powiadomienie innych zainteresowanych węzłów o dokonanych zmianach

Odporność na awarie

Serwer rejestruje stan klientów

- Informacja o pobranych kopiach plików
- Komunikaty unieważniające

Po awarii

- Klient: kontrola ważności przechowywanych kopii plików

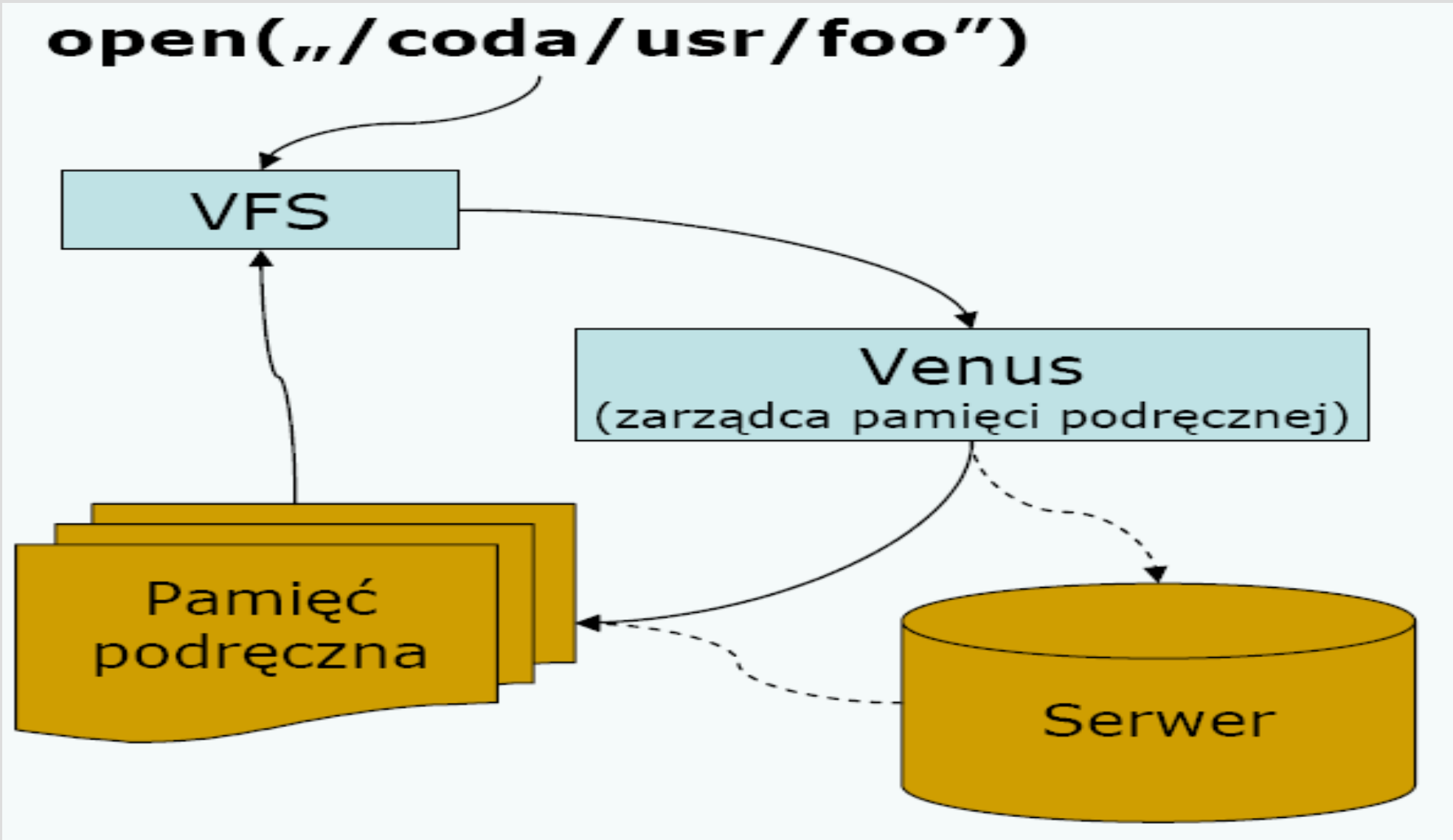
Automatyczne unieważnianie replik

- Brak informacji od serwera przez okres czasu T (typowo kilka minut) powoduje unieważnienie kopii

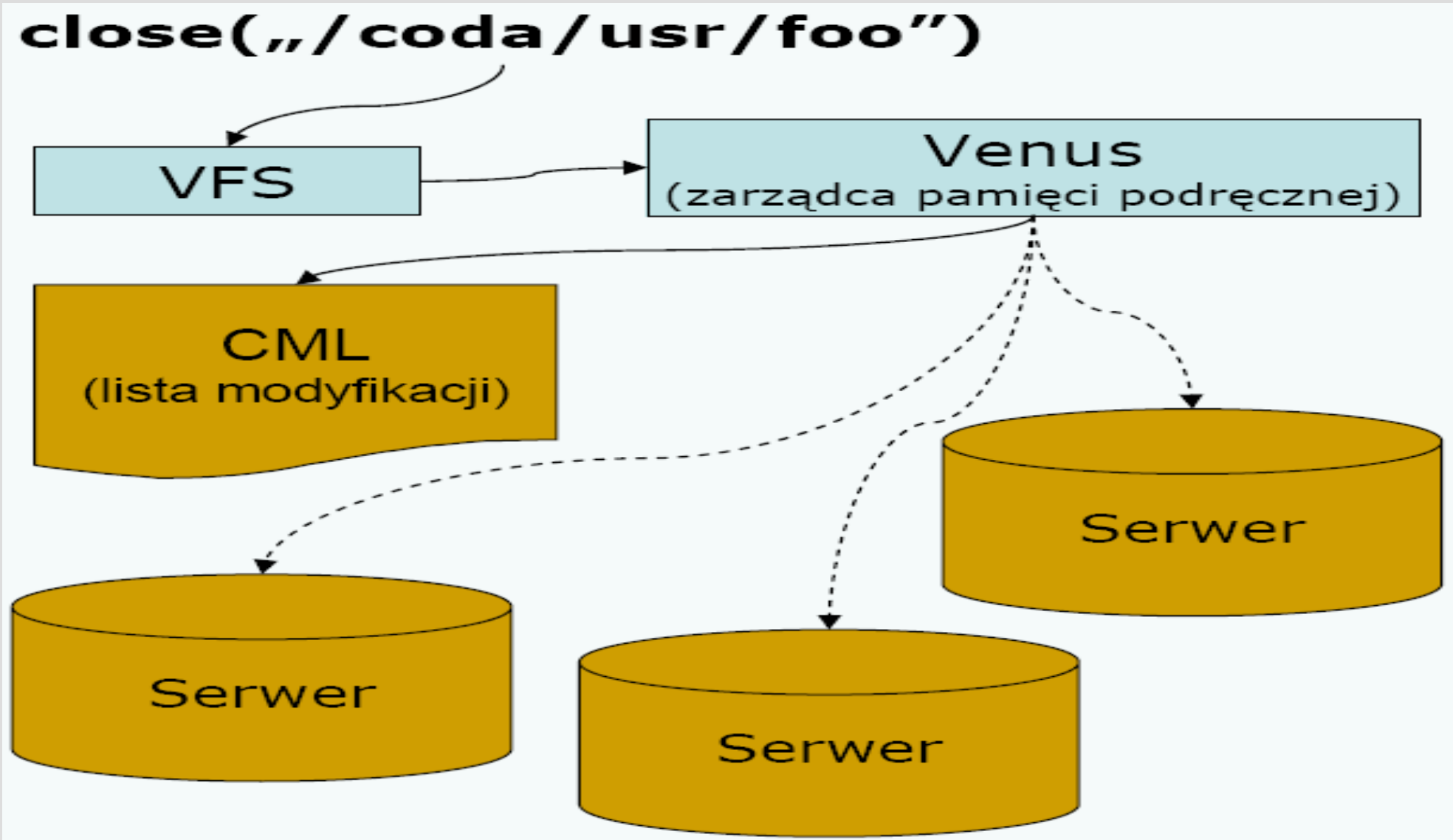
Coda

- Replikacja plików modyfikowanych
- Lepsze tolerowanie uszkodzeń
- Dostępność zasobów przy braku dostępu do serwerów
- Skalowalność i emulacja uniksowej semantyki dostępu do plików jak w AFS

open


open(„/coda/usr/foo”)
VFS
Venus
(zarządca pamięci podręcznej)
Pamięć
podręczna
Serwer

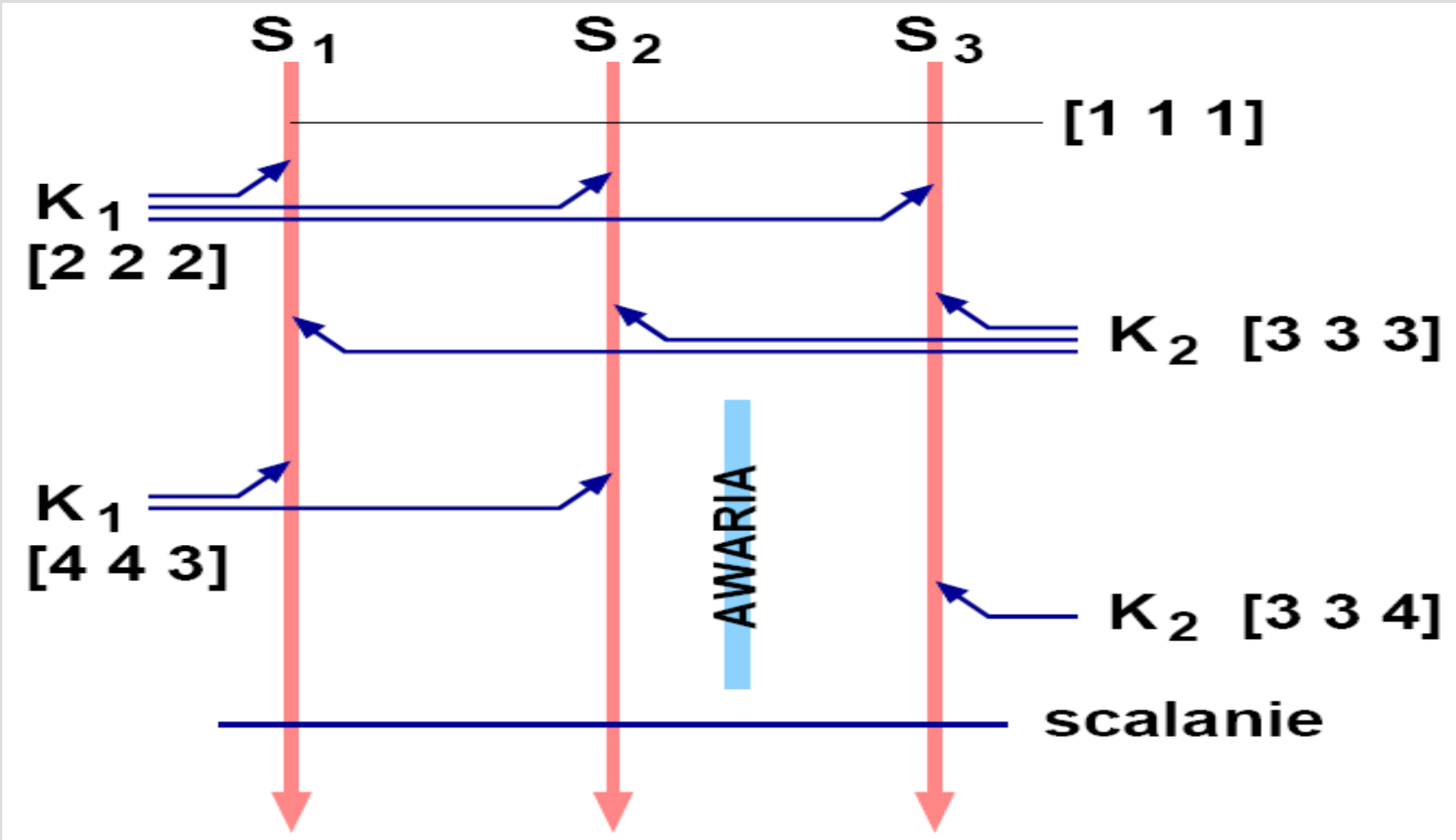
close


close(„/coda/usr/foo”)
VFS
Venus
(zarządca pamięci podręcznej)
CML
(lista modyfikacji)
Serwer
Serwer
Serwer

Spójność danych

- Wektory wersji: Coda Version Vector (CVV)
- Liczniki modyfikacji pliku na poszczególnych serwerach
- Użytkownik musi ręcznie rozwiązać konflikty powstające podczas operacji scalania

Konflikt podczas scalania



GoogleFS

- Mocno dedykowany
- Stworzony z założeniem, że błędy są czymś naturalnym
- Oferuje replikacje
- Przezroczystość współbieżności

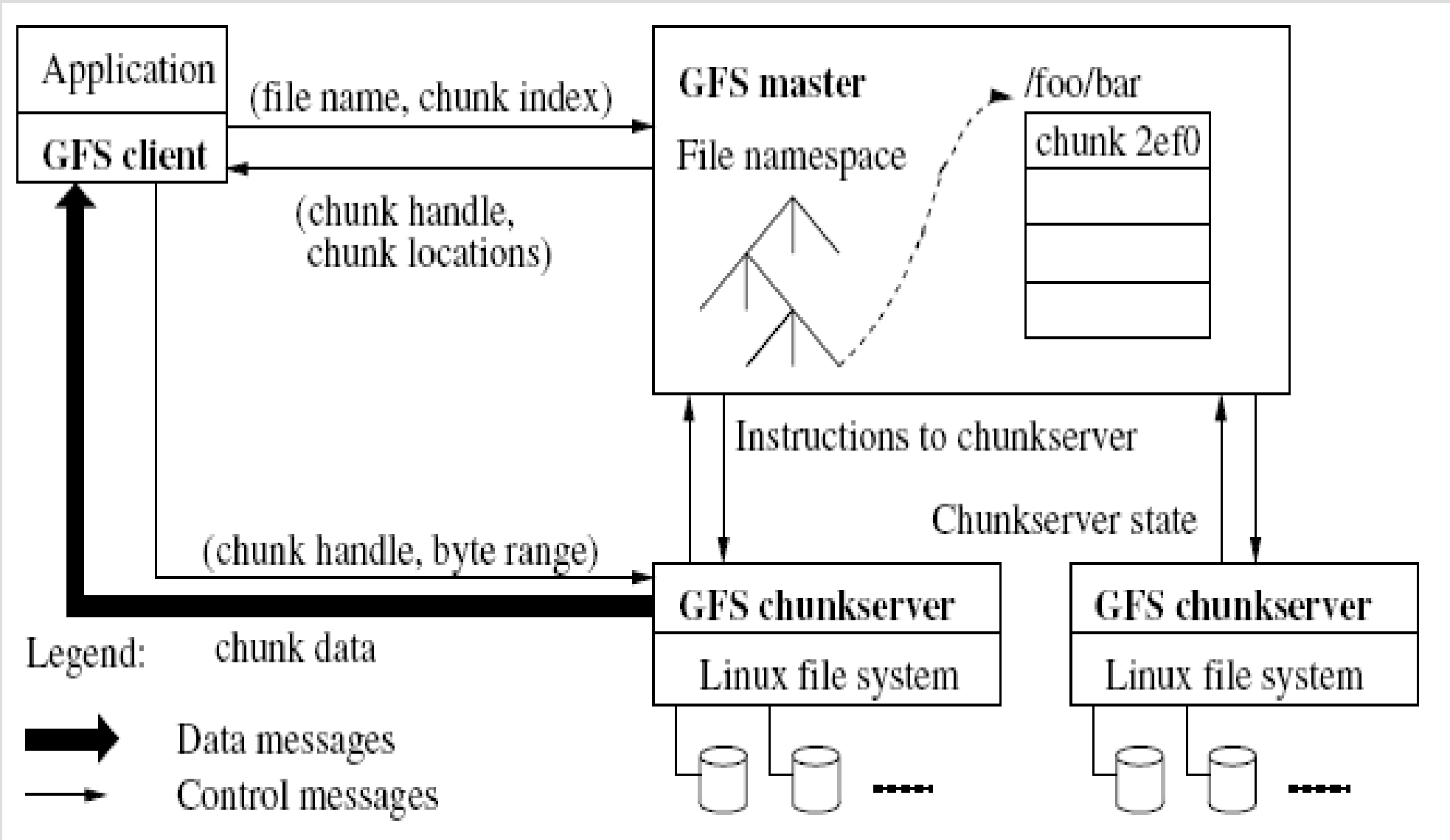
Założenia

- Dużo taniego hardware'u
- Większość plików >100MB. Pliki po kilka GB są powszechne. Małe pliki występują, ale dostęp nie jest optymalizowany
- Czytanie dużymi strumieniami lub małe losowe odczyty
- Zapis zazwyczaj jako dodawanie na koniec
- Potrzebne wsparcie dla problemu producent-konsument

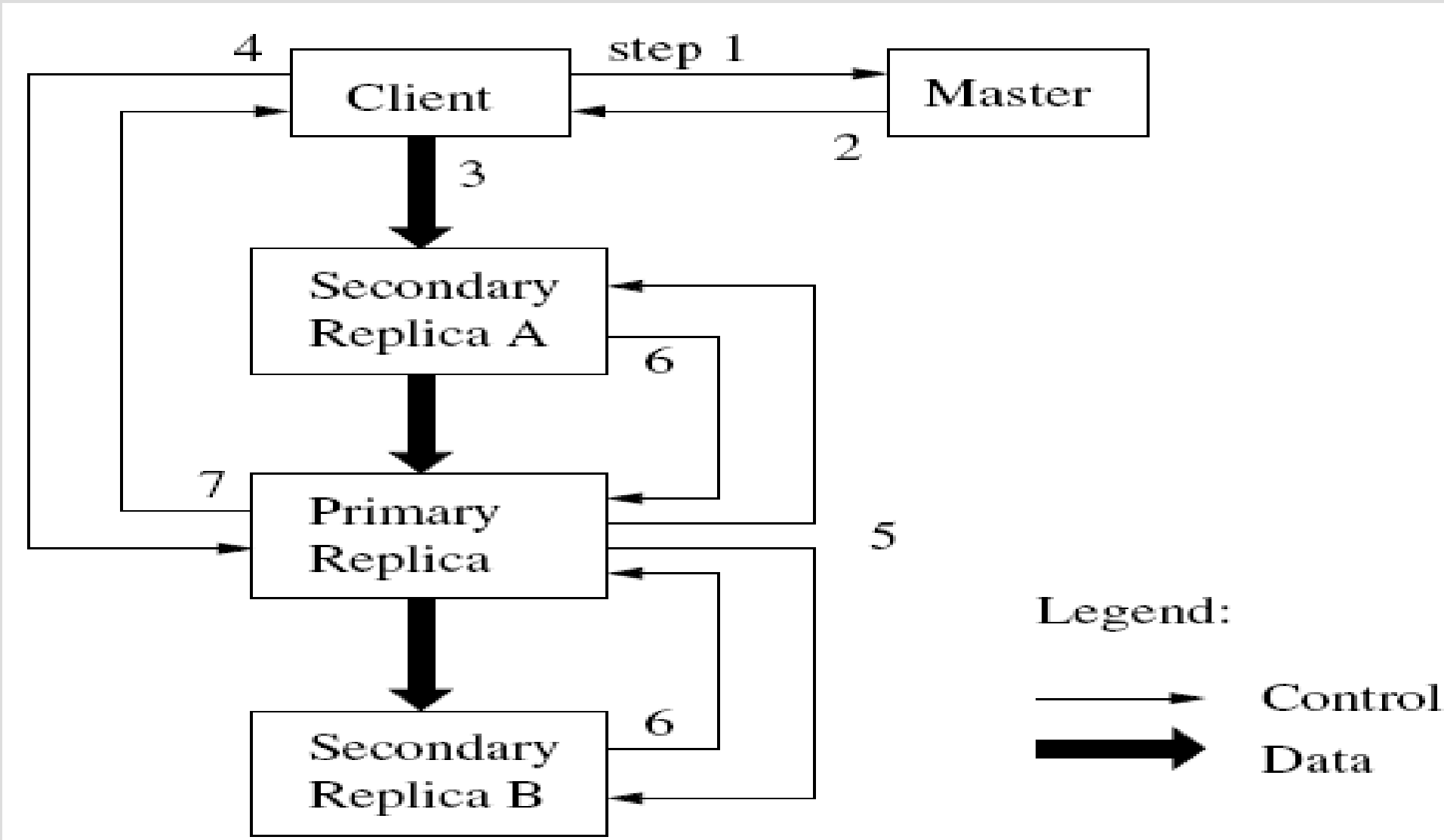
Architektura

- Jeden serwer z metadanymi – master
- Wiele maszyn z danymi – chunkservers
- Plik jest dzielony na kawałki (chunks) o stałym rozmiarze
- Każdy kawałek jest replikowany stałą liczbę razy
- Master synchronizuje globalnie cały klaster periodycznymi wiadomościami HeartBeat

Odczyt


Application
GFS client
(file name, chunk index)
(chunk handle,
chunk locations)
GFS master
File namespace
/foo/bar
chunk 2ef0
Instructions to chunkserver
Chunkserver state
(chunk handle, byte range)
GFS chunkserver
Linux file system
GFS chunkserver
Linux file system
Legend:
chunk data
Data messages
Control messages

Zapis


4
step 1
Client
Master
2
3
Secondary
Replica A
6
7
Primary
Replica
5
6
Secondary
Replica B
Legend:
Control
Data

Odśmiecanie

- Usunięte pliki nie są natychmiast usuwane
- Sierotki (orphan chunks) są eliminowane :) w regularnych odstępach czasu

ZFS (Zettabyte File System)

- The last word in file systems
- Otwarty kod źródłowy
- 584 pliki
- 92.000 linii kodu
- 56 patentów (!)
- 5 lat pracy
- Oczywiście za darmo

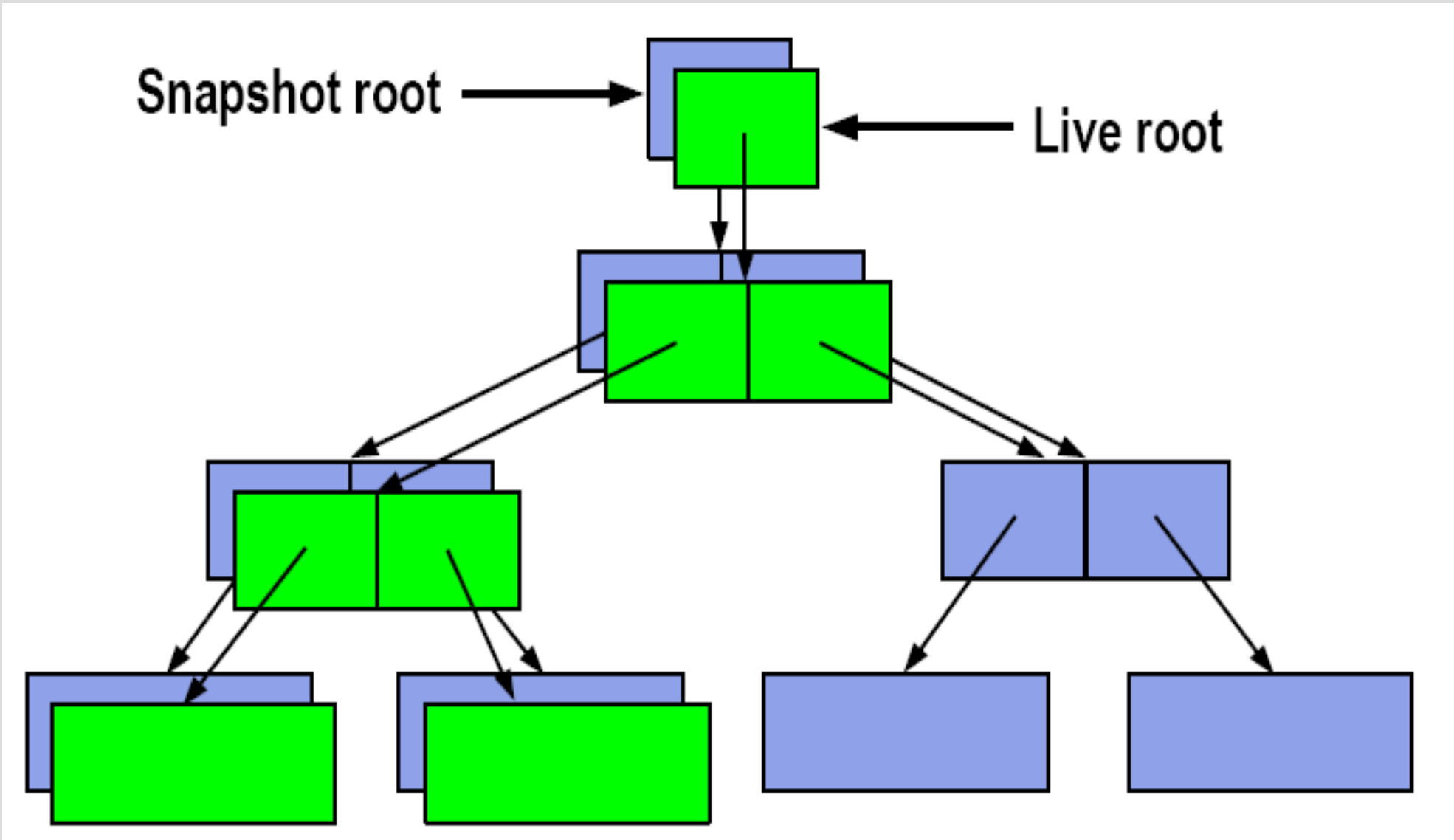
Podstawowe cechy

- System plików 128-bitowy
- System plików i manager wolumenów w jednym
- Zapewnienie integralności danych end-to-end
- Zabezpieczenie każdego bloku danych przez 256-bitową sumę kontrolną
- Data self-healing
- Model zapisów Copy-On-Write
- Scrubber sprawdzający automatycznie lub na żądanie konsystencję danych

Podstawowe cechy (2)

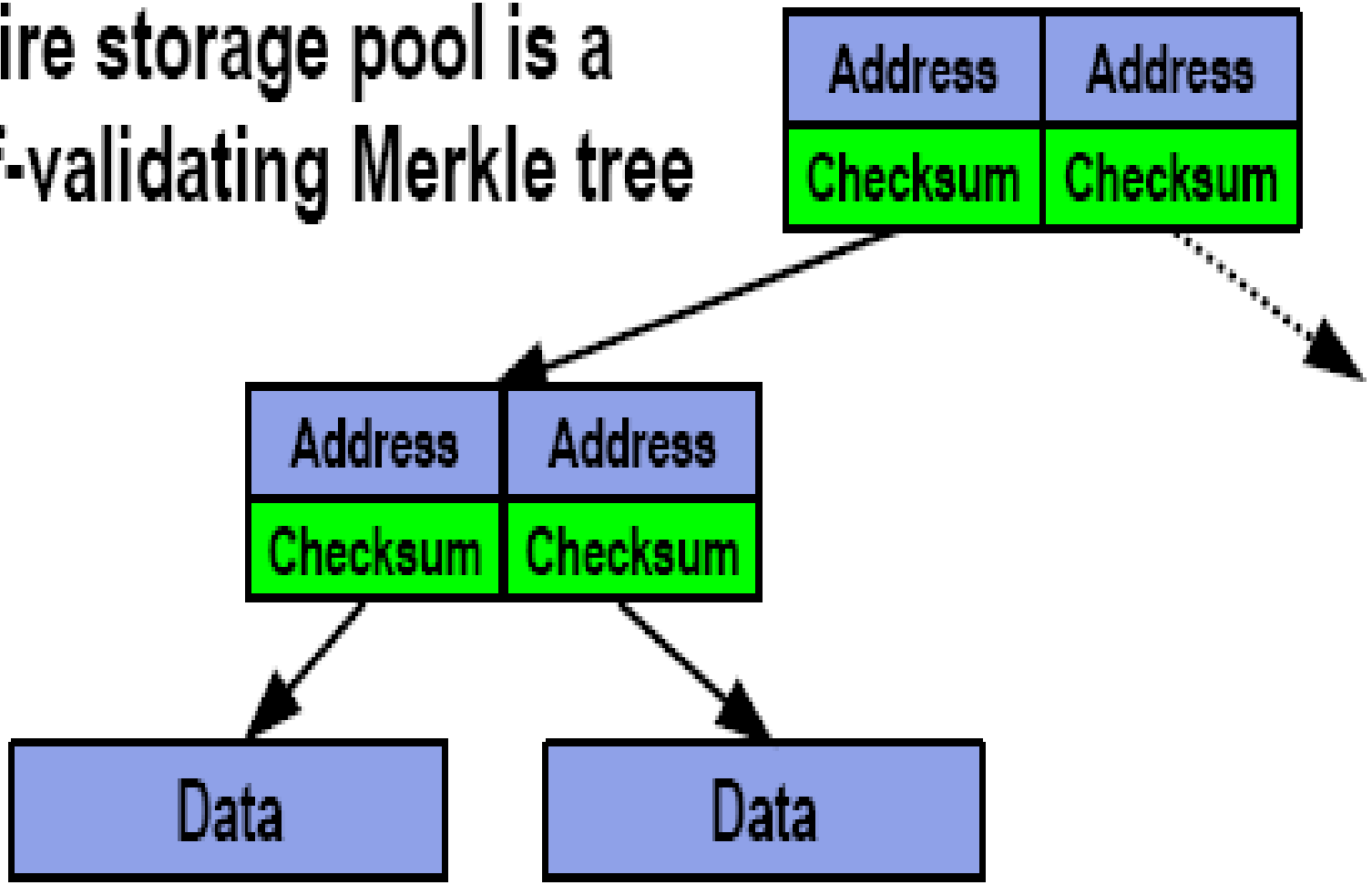
- Zmienny rozmiar bloku danych (od 512B do 128KB)
- Kompresja danych na poziomie bloku
- Snapshot, clone, rollback
- Listy kontroli dostępu
- Niskopoziomowy backup/restore, pełny lub inkrementacyjny
- Integracja z NFS
- Niezależność od platformy sprzętowej

COW, snapshoty

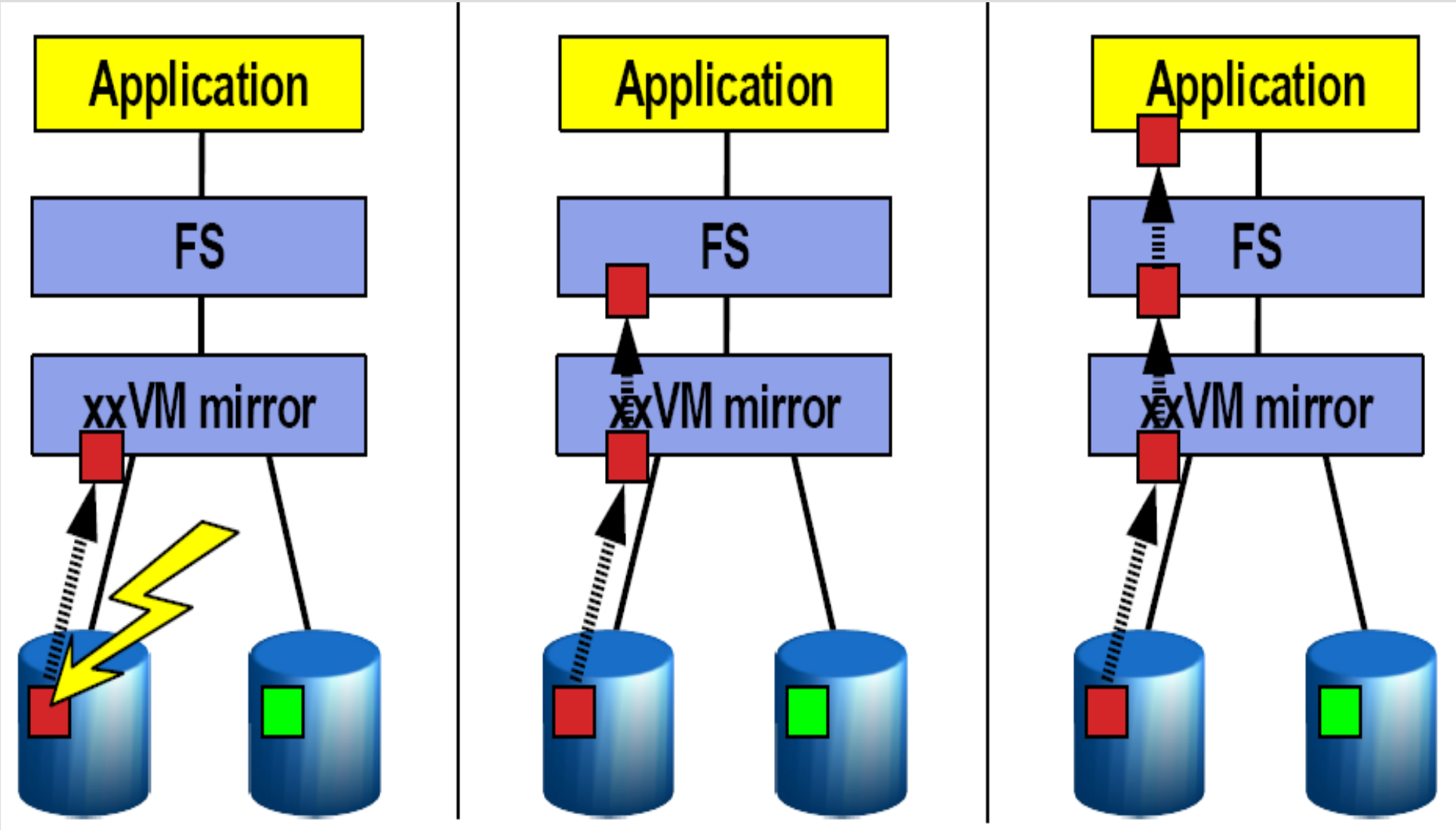


Walidacja ścieżki

- Entire storage pool is a self-validating Merkle tree



Automatyczna naprawa danych



Automatyczna naprawa danych