

Dynamiczna Waga Kolejowa - Powiśle



Konrad Chodyna

Paweł Janiak

Marcin Stasiuk

Plan prezentacji



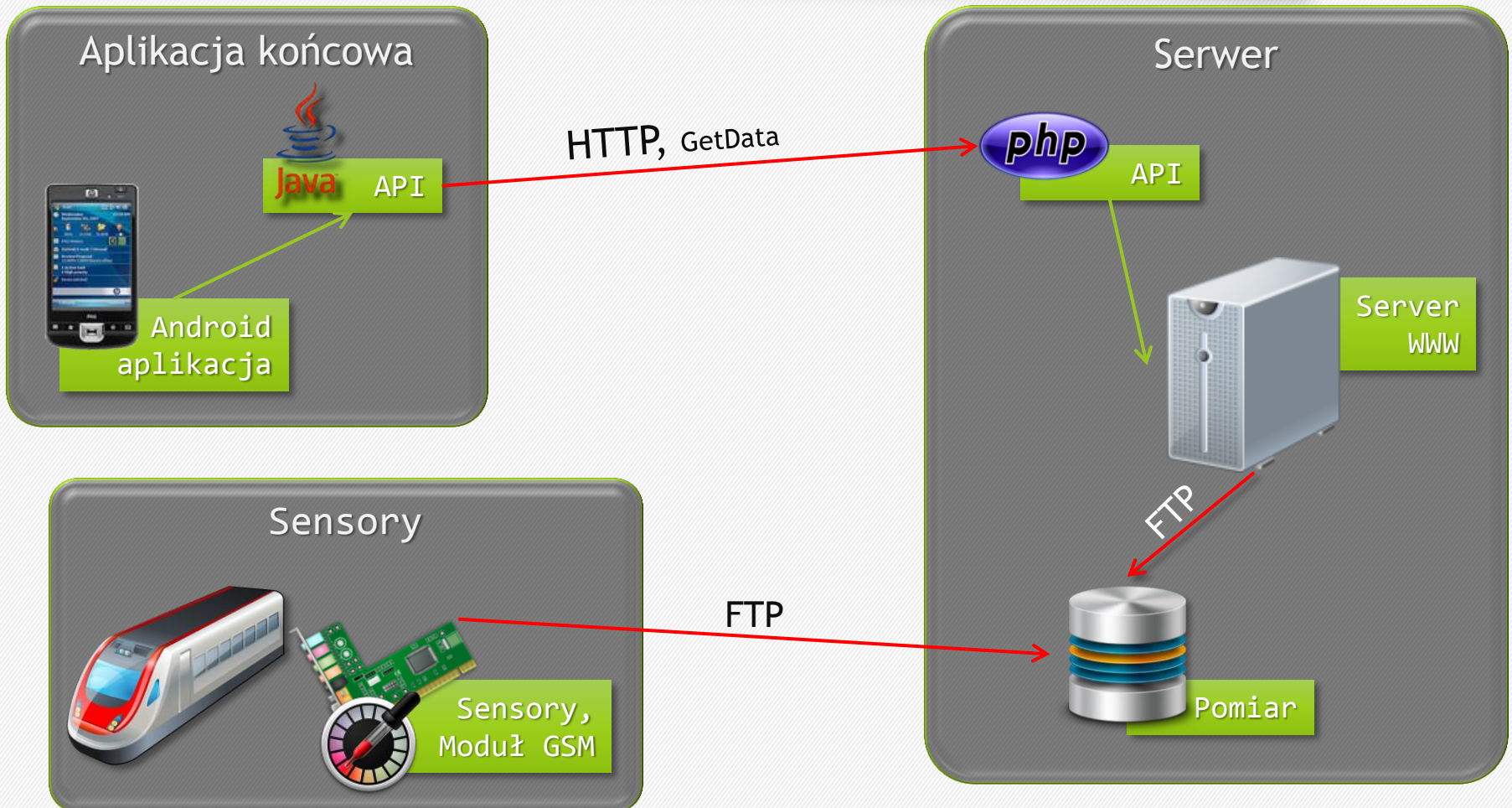
2

- Omówienie architektury systemu
 - Sensory
 - Serwer
 - Aplikacja mobilna
 - Prezentacja „na żywo” działania aplikacji

Architektura systemu



3



Elementy wchodzące w skład systemu pomiarowego



4

- Sensors - piezoelectric patches
- Analog Input/Output Module
- Controller - NI 9075
- Amplifier - 8 channel, conditional amplifier with input $\pm 5\text{ V}$
- GSM Module - SEA 9721

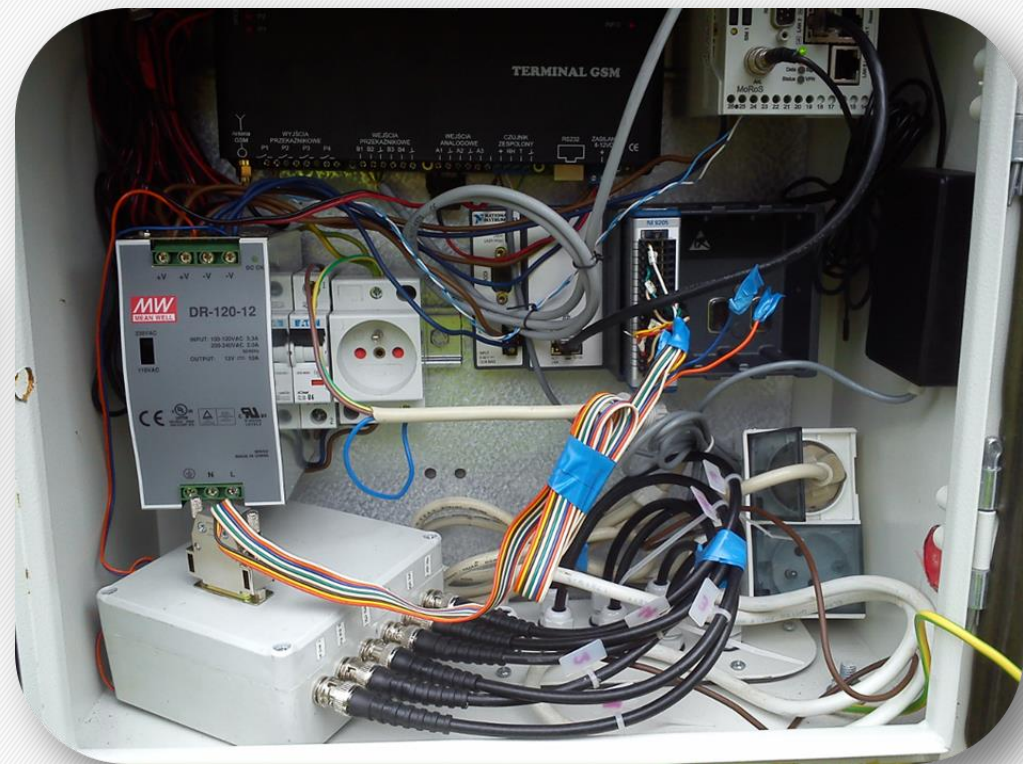


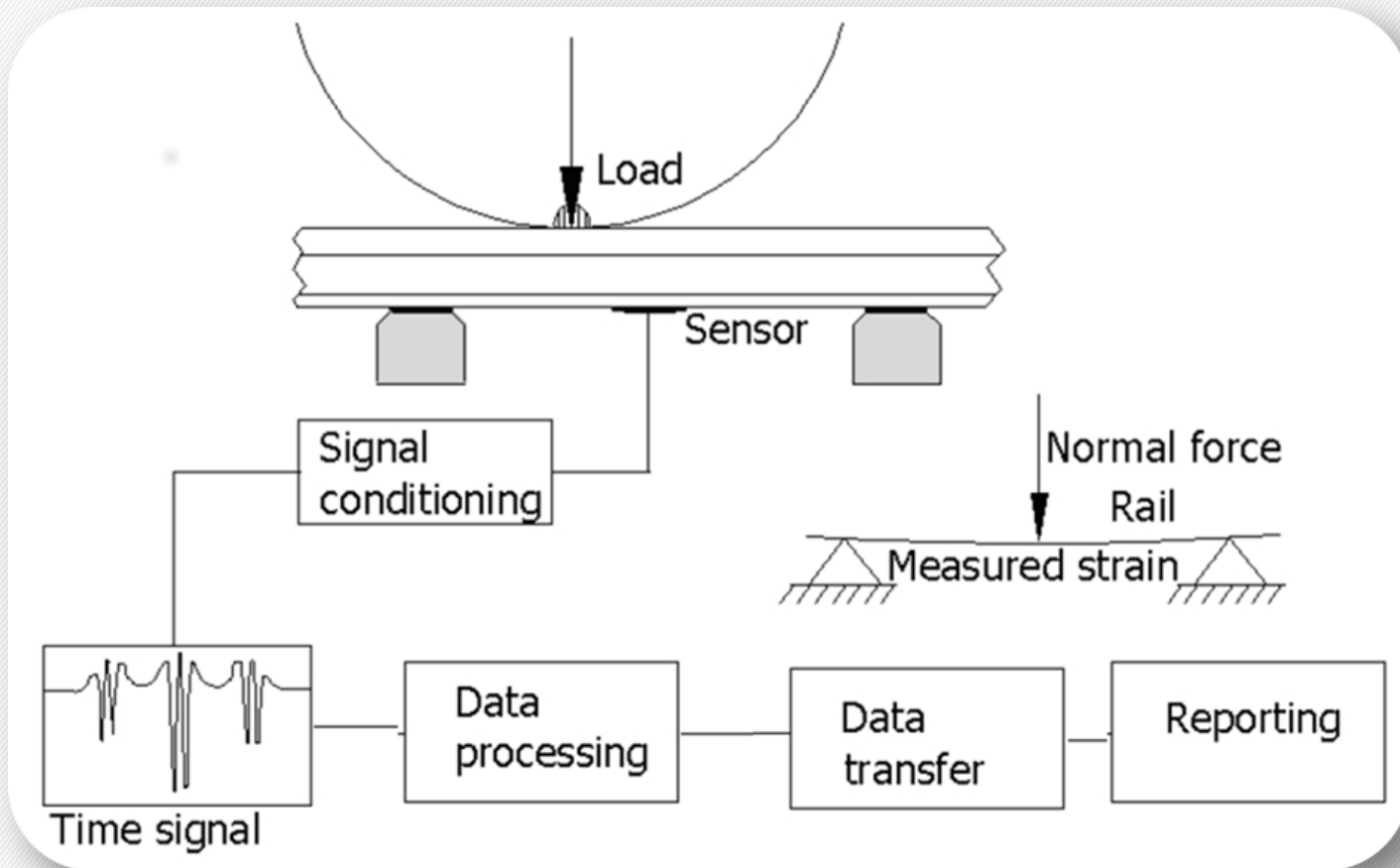
Stanowisko pomiarowe



5

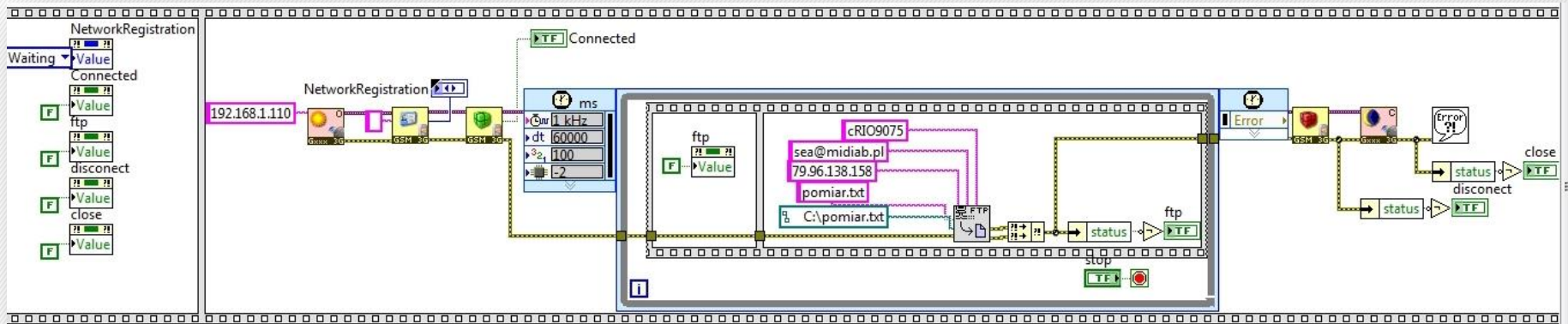
- Controller
cRIO controller NI 9075







- Program wysyłający plik pomiarowy na serwer ftp



Lokalizacja instalacji dynamicznej wagi drogowej



8



Rys.1 Lokalizacja instalacji Dynamicznej Wagi Drogowej



- Ograniczony dostęp i możliwości serwera źródłowego
- Wprowadzenie serwera pośredniczącego
 - cron
 - komunikacja z serwerem źródłowym poprzez ftp
 - możliwość przechowywania wszystkich pomiarów
 - umieszczanie skryptów
 - backup danych
 - udostępnianie pomiarów
 - web api



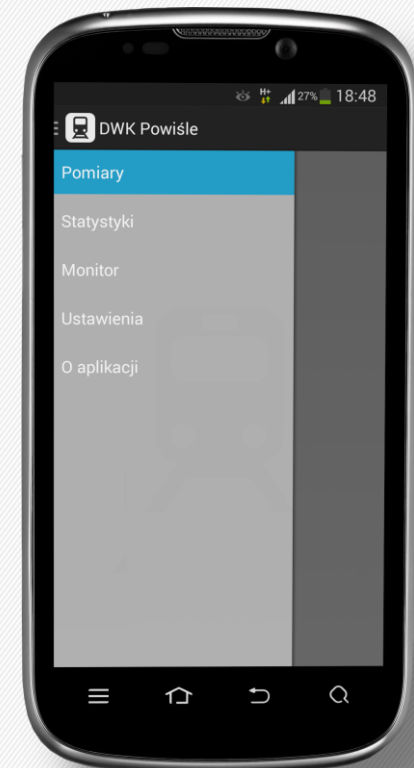
- obecnie jedna metoda - *GetData*
- dane zwracane w formacie JSON
- parametry metody, wartości domyślne
 - *from* [0]
 - *to* [INT_MAX]
 - *limit* [INT_MAX]
 - *offset* [0]
 - *order* [asc]
- wszystkie parametry można ze sobą łączyć; kolejność oraz wielkość liter nie ma znaczenia
- przykłady wywołań
 - Pobierz wszystkie rekordy:
GetData
 - Pobierz ostatni pomiar:
GetData?limit=1&order=desc
 - Pobierz 10 pierwszych pomiarów z pominięciem pierwszego:
GetData?offset=1&limit=10

Aplikacja końcowa



11

- Wybór platformy systemowej
- Komunikacja aplikacji z serwerem
- Dane przechowywane lokalnie na urządzeniu mobilnym
- Prezentacja działania oraz omówienie funkcjonalności aplikacji



Aplikacja końcowa



12

- Platforma - Android (min. SDK: 9, doc. SDK: 21)
- Wymagany dostęp do Internetu
- Wykorzystane biblioteki:
 - *DWK Java API*
 - *MPAndroidChart*

