



# [DOC] Bramki Komunikacyjne AGF - dokumentacja techniczna

Created by Michał MM. Mewald, last modified by Tomasz TS. Siemiątkowski on Jan 16, 2018

## Bramki Komunikacyjne AGF

- Bramki Komunikacyjne AGF
  - Ogólny sposób działania bramek AGF
  - Współpraca z cache
  - Bramki w managerach i poza nimi. Logowanie.
  - Udział Spring Framework w konfiguracji bramek.
  - Wspólne parametry konfiguracyjne wszystkich bramek AGF
  - AGF SMPP
  - AGF SEMA
  - AGF CIMD
  - AGF UCP
  - Cykl życia ramki
    - T-mobile
      - sms MO / IN / operator -> avantis
      - sms MT / OUT / avantis -> operator
    - ORANGE
      - sms MO / IN / operator -> avantis
      - sms MT / OUT / avantis -> operator
    - PLUS
      - sms MO / IN / operator -> avantis
      - sms MT / OUT / avantis -> operator
    - PLAY
      - sms MO / IN / operator -> avantis
      - sms MT / OUT / avantis -> operator
  - Sposoby badania ruchu na bramkach
    - Sprawdzanie na bazie
    - Sprawdzanie w logach bramek za pomocą przykładowego tail/cat/grep'a.
      - ERA / PLUS / ORANGE / PLAY / CYFROWY POLSAT / SILVERSTREET / CARDBOARDFISH
    - Czas ważności wiadomości
    - Logi:
  - MONITORING
    - AGF Gateways errors
      - Szukane błędy:
      - Częstotliwość:
      - Weryfikacja:
      - Reakcja:
    - PLAY-MT BLEDY
      - Weryfikacja:
      - Reakcja:

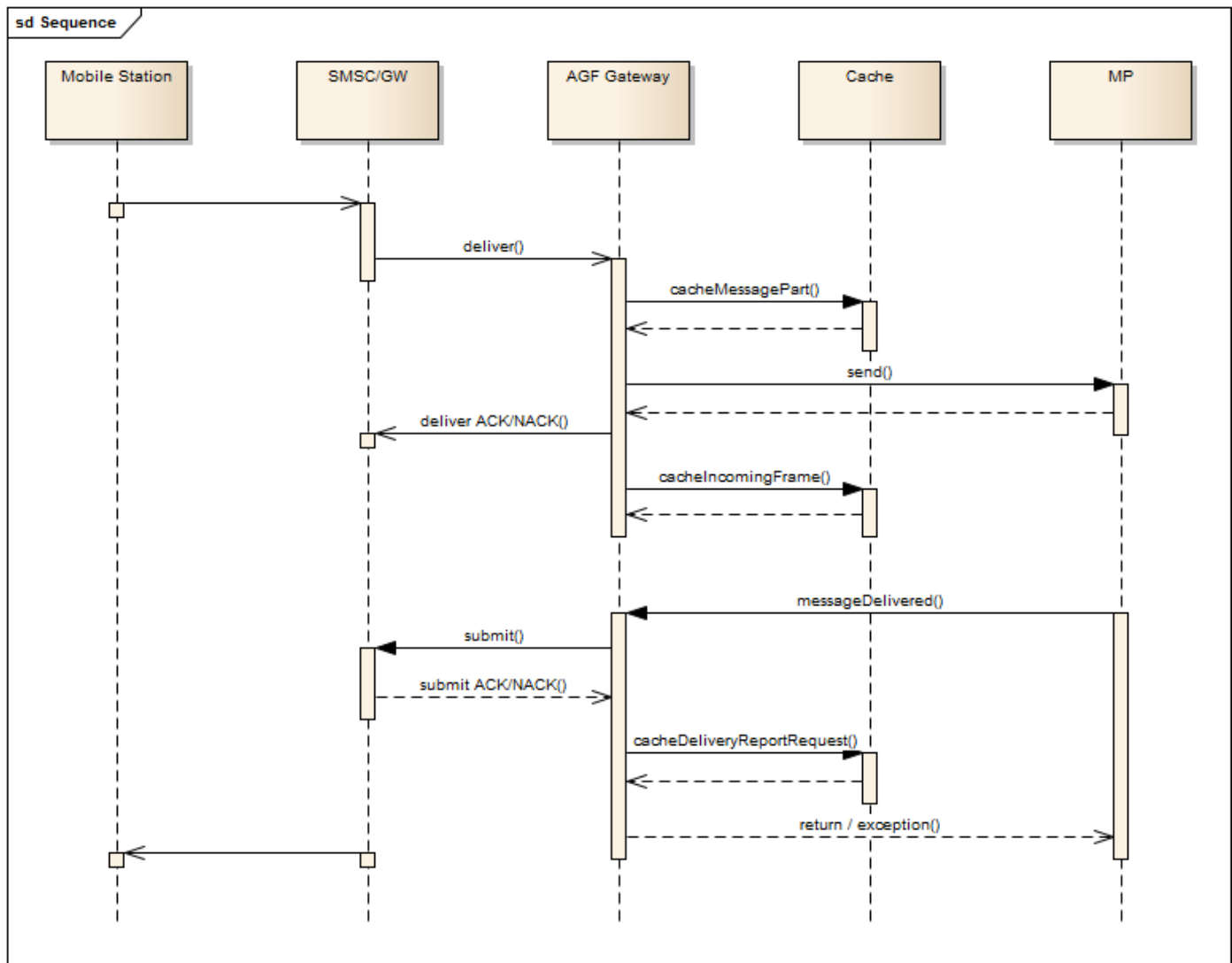
W pierwszych słowach mojego listu donoszę, że powoli dobiega końca proces wdrażania bramek komunikacyjnych AGF dla wszystkich polskich oraz wybranych zagranicznych operatorów. Oznacza to, że z grubsza od końca marca 2012 większość bramek komunikacyjnych w Avantis jest skonstruowana jako nadbudowa na wspólną bibliotekę (=async-gateway-framework-core=), a co za tym idzie - wszystkie, przynajmniej z założenia, działają w identyczny sposób. Sposób ten będzie przedmiotem moich egzaltacji w niniejszym dokumencie, który warto przeczytać, przyswoić i recytować na wyrywki.

## Ogólny sposób działania bramek AGF

Bramki AGF wykorzystują do komunikacji sieciowej asynchroniczny mechanizm JBoss Netty; asynchroniczny w tym wypadku oznacza dla nas dwie rzeczy:

- Przy wysyłaniu danych do sieci, Netty zwraca nam kontrolę zanim dane zostaną faktycznie wysłane
- Przy odbieraniu danych nie oczekujemy na nie w blokującej pętli, Netty przekazuje nam je tak szybko, jak się pojawiają.

Zaletą takiego układu jest to, że znacznie trudniej bramkę zakleszczyć, wadą - że znacznie trudniej powiązać wysyłaną ramkę z jej potwierdzeniem. Ale o tym to może za chwilę :)



Obrazek przedstawia diagramy sekwencji dla wiadomości wysyłanych z telefonów (MO) oraz wysyłanych na telefon (MT). Dla MO sprawa wygląda stosunkowo prosto - ze strony SMSC operatora wpada do nas ramka z SMSem użytkownika (w niemal wszystkich protokołach jej nazwa ma coś wspólnego z **DELIVER**). My podejmujemy następujące kroki:

- Sprawdzamy, czy ramka nie jest czasem częścią SMSa wieloczęściowego - jeżeli tak, cache'ujemy i potwierdzamy.
- Jeżeli ramka jest SMSem jednoczęściowym albo ostatnią częścią wieloczęściowego, składamy go do kupy
- Wrzucamy powstałego SMSa do MP
- Jak się uda, cache'ujemy ramkę jako ramkę wchodzącą (aby móc uniknąć powtórzeń ze strony operatora) i potwierdzamy pozytywnie (ACK)
- Jak się nie uda, potwierdzamy negatywnie (NACK) i liczymy na to, że operator będzie tą wiadomość retransmitował.

Dla MT sprawa jest nieco bardziej zawiła:

- Jeżeli na bramkę wpada SMS wieloczęściowy, szatkujemy go na odpowiednie porcje
- Każdą porcję formujemy w ramkę wychodzącą (w niemal wszystkich protokołach jej nazwa ma coś wspólnego z **SUBMIT**)
- Wysyłamy ramkę (lub, jeżeli SMS jest wieloczęściowy, wiele ramek w pętli, w jednym wątku) do operatora, zapamiętując sobie (w pamięci RAM) informację pozwalającą na zidentyfikowanie potwierdzenia (potwierdzeń), które nadejdzie ze strony operatora asynchronicznie
- **Zatrzymujemy wątek, który wysłał ramkę/ramki (jest to wątek MP) aż do momentu nadejścia wszystkich potwierdzeń (lub timeoutu).**
- **Jeżeli potwierdzenie ze strony operatora jest negatywne, albo nie dostaliśmy ich na czas, rzucamy błędem i wiadomość wraca do MP**
- Jeżeli potwierdzenie było pozytywne, wysyłamy do MP potwierdzenie doręczenia do SMSC (jeżeli było zamawiane), cache'ujemy żądanie doręczenia do użytkownika (jeżeli było zamawiane) i bezbłędnie kończymy metodę `messageDelivered`

Bardzo ważne są punkty, które zaznaczyłem na tłusto, bo są one teraz esencją konsystencji działania wszystkich bramek. Otóż, kiedyś-kiedyś było tak, że bramki działały w trybie send-and-wait, wysyłały jednego SMSa, czekały na potwierdzenie i wysyłały następnego. Później nadeszła era windowingu - bramka mogła wysłać kilka-kilkanaście SMSów (parametr określający, ile tych SMSów może być, nazywany jest **szerokością okna**) nie czekając na potwierdzenia ze strony SMSC. Działało to znacznie szybciej, no ale miało jedną zasadniczą wadę - już po wysłaniu ramki typu **submit** do SMSC uznawaliśmy SMSa za poprawnie wysłanego - a gdyby coś się poniszczyło (np. timeout), bramka sama musiała się martwić o retransmisję, a usługa źródłowa nawet nie wiedziała o tym, że może być problem.

I tak zrodził się pomysł pośredni, wykorzystujący windowing, lecz pozostający pod kontrolą MP w przypadku błędów. A mianowicie, musimy pozwolić bramce na to, żeby dostawała wiadomości równolegle w wielu wątkach (maks. w tyłu, ile wynosi szerokość okna), a każdy z wątków będzie działał w trybie send-and-wait. W ten sposób, z punktu widzenia operatora, nie czekamy na potwierdzenia z wysyłaniem następnych wiadomości - a jeżeli którykolwiek wątków się wywali (albo wytimeoutuje), wiadomość wróci do MP i zostanie zaraz zretransmitowana.

Jeżeli ktoś jest ciekawy, jakim cudem można zablokować wątek (czyli de facto uczynić komunikację synchroniczną), kiedy silnik pod spodem jest asynchroniczny... cóż, łatwiej kijek pocienkować, niż go później pogrubiasić ;)

Tutaj warto wspomnieć o zależnościach szerokości okna, liczby połączeń i parametru max-messages w konfiguracji MP-Service'u. Z opisu powyżej wynika, że bramka może obsłużyć równolegle tyle wiadomości, ile wynosi suma szerokości okien na wszystkich połączeniach, jakie ustanawia do SMSC. Jeżeli więc ma ona np. 2 równoległe połączenia (parametr poolSize, często w bramkach SMPP), a na każdym z nich okno ma szerokość 10 (parametr windowSize=), to bramka per saldo jest w stanie obsłużyć 20 wiadomości równolegle - i na taką wartość trzeba ustawić parametr =max-messages.

## Współpraca z cache

Jak można się zorientować z diagramów sekwencji - i elaboratu w powyższych akapitach - kluczowym dla działania bramek jest działanie cache'a. Bez niego wszystko leży i kwiczy. Cache jest używany w czterech miejscach:

- do przetrzymywania przychodzących części SMSów wieloczęściowych (cache trwałe, TTL elementów domyślnie 1 doba)
- do przetrzymywania ramek przychodzących w celu wykrywania retransmisji operatora (cache replikowany, TTL elementów domyślnie 1 doba)
- do przetrzymywania wyników wysyłania ramek SMSów wieloczęściowych (cache trwałe, TTL elementów domyślnie 1 doba)
- do przetrzymywania żądań raportów doręczeń (cache trwałe, TTL elementów domyślnie 8 dób)

Druga z wymienionych operacji jest przeprowadzana dla każdej, każddziusieńkiej ramki przychodzącej, więc jeżeli coś będzie nie halo z cache, nie wejdzie do nas żadna wiadomość MO. Klops!

## Bramki w managerach i poza nimi. Logowanie.

Niektóre z bramek (SMPP i SEMA) z reguły obsługują wiele numerów premium na pojedynczym koncie w SMSC GW. Tego typu bramki z reguły uruchamia się *sauté* - jedna usługa MP, jeden log, jeden problem (choćby potencjalnie wiele połączeń, bo niektórzy operatorzy, np. Play, dopuszczają wielokrotne podłączenie się do jednego portu).

Nieco wyższym stopniem wtajemniczenia są bramki pracujące pod kontrolą managerów - dodatkowych aplikacji uruchamiających wiele (rekordzistą w tym względzie jest główny manager Orange - zarządza ok. 180 bramkami ;) ) bramek równolegle. Zwyczajowo, pod kontrolą managerów uruchamia się bramki Orange (protokół UCP) i Plus (protokół CIMD), z racji faktu, że jedna bramka obsługuje tylko jeden numer premium - i konia z rzędem temu, komu chciałoby się odpalać to z palca ;). W zasadzie jest to tylko pomoc przy uruchamianiu / zatrzymywaniu, ponieważ jako serwisy MP, podległe managerom, bramki działają zupełnie tak samo, jakby były uruchomione niezależnie od nich (istnieje również możliwość uruchomienia pojedynczej bramki UCP i CIMD, jeżeli zachodzi taka konieczność).

Przed erą AGF mieliśmy z managerami tego typu problem, że tylko część logów danej bramki lądowała w logu jej przynależnym, a reszta - w logu managera. Dlaczego tak się działo - to długa historia (w skrócie - konfiguracja logowania pozwalała na przekierowanie do logu wpisów tylko z głównej klasy bramki, a ze wszystkich klas pomocniczych już nie). Na potrzeby AGF udało się opracować taki mechanizm (wyższa szkoła jazdy, niem. *Hochfahrschule*), który poprawnie zapisuje do logów wszystko to, co tyczy się jednej bramki jako całości, nie tylko jej głównej klasy. W ten sposób, logi bramek uruchamianych pod managerem i poza nim powinny nieść identyczną informację.

Jeżeli już chodzi o logowanie, najważniejsze rzeczy dziejące się w bramkach są logowane tak samo dla wszystkich protokołów. Warto zapamiętać zatem takie kilka wyrażen dla grepa:

```
MP -> SMSC
SMSC -> MP
Raw frame being read
Read frame parsed into
Frame being sent
Raw frame being sent
Matched response.*with request
```

Pozwalających wyłowić, odpowiednio:

- Wysyłanie wiadomości do SMSC (MT)
- Włożenie do MP wiadomości od SMSC (MO/DR)
- Przeczytanie surowej ramki z sieci (surowa w sensie - tablica bajtów albo string)
- Przeparsowanie surowej ramki przychodzącej
- Próbę wysłania określonej ramki
- Próbę wysłania określonej ramki w postaci surowej
- Nadejście potwierdzenia dla wcześniej wysłanej do SMSC ramki (MT)

Warto również wspomnieć, że przy okazji przesadzania bramek na implementacje AGF pozbyliśmy się z konfiguracji logów sztywno zapisanej ścieżki /usr/local/logs/mpservices/comm. W tym miejscu figuruje teraz zmienna środowiskowa LOG\_PATH, którą można ustawić na starcie bramki / managera, i w ten łatwy sposób przekierować logi do innego katalogu. Albo do /dev/null ;).

## Udział Spring Framework w konfiguracji bramek.

Jak w większości aplikacji pisanych w Avantis, bramki AGF wykorzystują bibliotekę Spring Framework do konfiguracji zarówno bramek, jak i managerów. Punktem wyjścia jest zawsze nazwa bramki albo managera na dysku - skrypty startowe są tak zrobione, że na podstawie tej nazwy szukają dla siebie plików konfiguracyjnych w podkatalogu classes. Skrypt dla pojedynczej bramki szuka plików

[nazwa].appCtx.xml oraz [nazwa].mpCtx.xml, skrypt dla managera zaś - pliku [nazwa].appCtx.xml i pliku szkieletowej konfiguracji bramki (domyślnie =gatewayCtx.xml=).

Najpierw powiem słowo o bramkach: plik [nazwa].mpCtx.xml, z konfigurowalnych opcji zawiera w sumie tylko i wyłącznie MPID usługi i nazwę hosta MP Core. Nieco więcej dzieje się w pliku [nazwa].appCtx.xml - tam zapisana jest konfiguracja konkretnej bramki (jako bean o nazwie config, mający różne opcje dla różnych protokołów) oraz konfiguracja jej podłączenia do cache. O konkretnych opcjach konfiguracyjnych konkretnych bramek napiszę w odpowiednich podrozdziałach niżej, należy jednak pamiętać o jednej ważnej rzeczy: **w celu zachowania wstecznej kompatybilności, bramki SEMA, CIMD i UCP na stracie parsują również swoją konfigurację MP-Service'u i nadpisują znalezionymi tam wartościami konfigurację przepisaną ze Springa**. Dlatego też nie należy się szczególnie dziwić (np. w bramkach UCP), że sekcja parametrów bramki jest w Springu pusta, a bramka nadal działa :)

Z kolei jeżeli idzie o managera, to w pliku [nazwa].appCtx.xml też nie ma zbyt wiele - MPID managera (ponownie, w celu zachowania wstecznej kompatybilności, listę obsługiwanych bramek manager pobiera z konfiguracji MP-Service'u) i, ewentualnie (p. managery Orange), zmienioną ścieżkę do szkieletowej konfiguracji bramki (we właściwości gatewayContextPath, jeżeli jest ona nieobecna, domyślnie, jak pamiętamy, ma wartość /gatewayCtx.xml, gdzie / oznacza główny katalog na zmiennej CLASSPATH - czyli z reguły =./classes=). Szkieletowa konfiguracja bramki natomiast zawiera te wartości, które dla pracujących pod managerem są wspólne - np. kod operatora, konfiguracja cache itd.

## Wspólne parametry konfiguracyjne wszystkich bramek AGF

W większości konfiguracji bramek występują takie parametry jak:

- smscHost - adres hosta SMSC
- smscPort - port na hostcie SMSC
- username - użytkownik
- password - hasło
- alivePeriod - interwał wysyłania ramki podtrzymującej połączenie, w ms.
- operatorCode - kod operatora wymuszanego przez bramkę
- poolSize - liczba równoległych połączeń do SMSC
- windowSize - szerokość okna na każdym z połączeń do SMSC
- incomingMessageProcessors - ilość wątków, którymi netty będzie zarządzał podczas procesowania wiadomości. (jeżeli brak == windowSize)
- monitor\_pools - flaga, która wskazuje czy połączenia mają być monitorowane przez wątek nadzorujący. (default = true)

## AGF SMPP

Bramki SMPP z reguły uruchamia się pojedynczo (obsługują wiele numerów na pojedynczych połączeniach). Z ciekawszych ustawień specyficznych dla SMPP mamy tu:

- timeZone - strefa czasowa SMSC - zmienia się ją, gdy jest inna od naszej (CET/CEST) - czyli jeżeli daty w raportach przychodzą poprzysuwane
- submitResponseIDParser - klasa parsująca numer ramki w jej potwierdzeniu
- reportIDParser - klasa parsująca numer ramki w raporcie doręczeń.

Dwa ostatnie ustawienia są dość ważne. Dlaczego? Otóż, numer ramki jest czymś, po czym łączymy raport z oryginalną wiadomością. Gdy przychodzi do nas potwierdzenie ramki (SubmitSMResp), niesie ono w sobie parametr messageId nadany przez SMSC - tyle że każde SMSC nadaje ten parametr inaczej. Czasami jest to liczba szesnastkowa, czasami jakiś tekstowy krzak... różnie. Ważne jest natomiast to, w jakiej postaci ten sam identyfikator przychodzi w raporcie doręczeń - zdarza się, że mimo tego, że w potwierdzeniu ramki przychodzi liczba szesnastkowa, w raporcie przychodzi ona zakodowana dziesiętnie. Czasami zdarza się, że w potwierdzeniu ramki i w raporcie przychodzi ten sam tekstowy krzak. A zdarza się też (na Słowacji oczywiście), że w potwierdzeniu ramki przychodzi liczba, a w raporcie - tekstowe krzaki :) Natomiast, dzięki klasom parsującym możemy wszystkie te sytuacje obsłużyć.

Przykładowe klasy to:

- pl.avantis.agf.smpp.FallThroughParser - nie parsuje identyfikatora w ogóle
- pl.avantis.agf.smpp.LongRadixParser - parsuje identyfikator jako liczbę całkowitą o podstawie podanej w parametrze - najczęściej 10 lub 16
- pl.avantis.agf.smpp.RegexIdentifierParser - wyciąga fragment wg. wyrażenia regularnego i parsuje jako liczbę o danej podstawie.

Najczęściej spotykane kombinacje to:

- FallThroughParser zarówno dla potwierdzeń ramek, jak i dla raportów (np. Silverstreet)
- LongRadixParser(16) zarówno potwierdzeń ramek, jak i dla raportów (np. Play n1-n3)
- LongRadixParser(16) dla potwierdzeń ramek i LongRadixParser(10) dla raportów (np. Play n4, Sonatel)

## AGF SEMA

Bramki SEMA z reguły są uruchamiane samodzielnie i obsługują wiele numerów premium. Mają one jedno specyficzne dla siebie ustawienie:

- retrieveRequestOriginatingAddress - numer premium używany przy podtrzymywaniu połączenia

Natomiast dość intensywnie korzystają z dwóch ustawienia wspólnych, mianowicie portu i hosta lokalnego:

- localhost - nazwa hosta lokalnego (albo IP interfejsu), do którego bramka ma się przywiązać
- localPort - port lokalny, do którego bramka ma się przywiązać.

W przypadku bramek SEMA podłączonych do T-Mobile PL, musimy wychodzić z określonym zewnętrznym adresem IP, a nie umiemy zorganizować sobie tego inaczej, niż przywiązanie do określonego portu lokalnego i zrobienie jakiś czarów na routerach brzegowych. Należy tu podkreślić, że z tego ustawienia mogą korzystać wszystkie bramki AGF, choć w praktyce korzystamy z niego tylko w przypadku SEMY.

## AGF CIMD

Bramka CIMD ma w sumie tylko jedno ważne ustawienie:

- la - obsługiwany numer premium (z reguły z konfiguracji MP-Service)

Warto też zwrócić uwagę, że bramki CIMD domyślnie nie wysyłają numeru LA jako nadawcy wiadomości - SMSC uzupełnia go na podstawie portu, do którego bramka jest podłączona (nie dotyczy to nadpisów, które są akurat umieszczane w ramach wychodzących).

## AGF UCP

Ze względu na fakt, że specyfikacja UCP jest pełna niedomówień i niedociągnięć, konieczne było wprowadzenie do jej konfiguracji parametrów, które umożliwiają jej współpracę z wieloma różnymi operatorami, mając jednak z tyłu głowy fakt, że musimy skupić się na obsłudze Orange PL. Parametry to:

- la - obsługiwany numer premium (z reguły z konfiguracji MP-Service)
- useKeepAlive - flaga, czy używać ramki O/31 do podtrzymania połączenia (powinna być ustawiona na true dla Orange)
- senderOverrideMode
  - tryb nadpisu (odpowiada ustawieniu alphanumericSenderMode w konfiguracji MP). Ustawiony na 1, powoduje zamianę LA źródłowego (np. wysyłamy wiadomość z LA 7036 przez bramkę 6830), ustawiony na 2 - wprowadzenie do ramki nadpisu alfanumerycznego.

Jest natomiast jeszcze jedna muka. Orange PL, z niewiadomych przyczyn, wymaga posługiwania się numerami MSISDN pozbawionymi prefiksu kraju (48) - stara bramka Orange po prostu przycinała numery. Tym niemniej, inni operatorzy mogą mieć inne wymagania dotyczące transformacji MSISDNów, więc nowa bramka ma taką furtkę. Gdyby jednak zaszła konieczność takiej zmiany, można ją stosunkowo łatwo wprowadzić do bramki. Jeżeli jest się developerem ;)

## Cykl życia ramki

### T-mobile

#### sms MO / IN / operator -> avantis

Przeczytanie surowej ramki z sieci (surowa w sensie - tablica bajtów albo string):

```
2012-04-15 00:00:04,655 DEBUG [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
2012-04-15 00:00:04,656 INFO [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
43836393238373930333600000008075469d0393516a531323034313530303030303435363132303431353030303030343
2012-04-15 00:00:04,656 DEBUG [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
303336000000008075469d0393516a531323034313530303030303435363132303431353030303030343536)
```

Przeparsowanie surowej ramki przychodzącej:

```
2012-04-15 00:00:04,656 INFO [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
pl.avantis.agf.sema.frames.Header [opref=-898791109, type=0, oper=9, size=65], destAddr=7655, smsc
456] on channel: [id: 0x53fd2626, /192.168.2.112:12051 => /83.220.97.177:12100]
2012-04-15 00:00:04,656 DEBUG [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
gf.sema.frames.Header [opref=-898791109, type=0, oper=9, size=65], destAddr=7655, smscRefNo=355088
2012-04-15 00:00:04,656 DEBUG [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
2012-04-15 00:00:04,668 DEBUG [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
cheService): totalTime=1645643, totalCount=105926
2012-04-15 00:00:04,668 DEBUG [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
2012-04-15 00:00:04,668 DEBUG [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
ngle part
2012-04-15 00:00:04,668 DEBUG [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
2012-04-15 00:00:04,670 DEBUG [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
2012-04-15 00:00:04,670 INFO [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
2012-04-15 00:00:04,670 DEBUG [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
```



**Stworzenie payload objectu:**

```
2012-04-15 00:00:04,670 INFO [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
e=Sun Apr 15 00:00:04 CEST 2012, externalSendDate=Sun Apr 15 00:00:04 CEST 2012, deliveryDate=null
payloadDirection=IN, text='TRANSFER', dataCodingScheme=0, partsNumber=1}
2012-04-15 00:00:04,671 INFO [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
lse
```

**Wysłanie wiadomości do MP:**

```
2012-04-15 00:00:04,740 INFO [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
```

**Zapisanie w cache'u:**

```
2012-04-15 00:00:04,740 DEBUG [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
2012-04-15 00:00:04,747 DEBUG [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
2012-04-15 00:00:04,747 INFO [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
```

**Wysłanie ack'a do operatora:**

```
2012-04-15 00:00:04,747 INFO [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
```

**Wysłanie ramki (ACK):**

```
2012-04-15 00:00:04,747 INFO [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
```

**Wysłanie ramki (ACK) w postaci surowej:**

```
2012-04-15 00:00:04,747 INFO [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
```

**sms MT / OUT / avantis -> operator****Odebranie wiadomości z MP:**

```
2012-04-15 00:00:06,094 INFO [RMI TCP Connection(940)-192.168.2.75] [SEMAGateway] [20482876584] M
2012-04-15 00:00:06,095 DEBUG [RMI TCP Connection(940)-192.168.2.75] [SEMAGateway] [20482876584] C
2012-04-15 00:00:06,095 INFO [RMI TCP Connection(940)-192.168.2.75] [SEMAGateway] [20482876584] M
```

**Sprawdzenie parametru " ValidityDate " (czas ważności wiadomości):**

```
2012-04-15 00:00:06,095 DEBUG [RMI TCP Connection(940)-192.168.2.75] [SEMARequestFactory] [2048287
```

**Przekwertowanie ramki i ustawienie kodowania:**

```
2012-04-15 00:00:06,095 INFO [RMI TCP Connection(940)-192.168.2.75] [SmsPduConverter] [2048287658
2012-04-15 00:00:06,095 INFO [RMI TCP Connection(940)-192.168.2.75] [SmsPduConverter] [2048287658
2012-04-15 00:00:06,095 DEBUG [RMI TCP Connection(940)-192.168.2.75] [SEMARequestFactory] [2048287
```

**Podzielenie wiadomości i wrzucenie jej do cache'a:**

```
2012-04-15 00:00:06,095 DEBUG [RMI TCP Connection(940)-192.168.2.75] [SEMAGateway] [20482876584] S
2012-04-15 00:00:06,095 DEBUG [RMI TCP Connection(940)-192.168.2.75] [RMICConnector] [20482876584]
2012-04-15 00:00:06,101 DEBUG [RMI TCP Connection(940)-192.168.2.75] [RMIClientInterceptorProxy] [
2012-04-15 00:00:06,101 DEBUG [RMI TCP Connection(940)-192.168.2.75] [SEMAAcknowledgeManager] [204
```

**Próba wysłania określonej ramki:**

```
2012-04-15 00:00:06,101 INFO [RMI TCP Connection(940)-192.168.2.75] [SEMAFrameEncoder] [204828765
```

**Próbie wysłania określonej ramki w postaci surowej:**

```
2012-04-15 00:00:06,101 INFO [RMI TCP Connection(940)-192.168.2.75] [SEMAFrameEncoder] [204828765
```

**Nadejście -> sparsowanie ramki ACK (z smsc operatora):**

```
2012-04-15 00:00:06,113 DEBUG [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
2012-04-15 00:00:06,113 INFO [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
2012-04-15 00:00:06,113 INFO [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
2012-04-15 00:00:06,113 INFO [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
2012-04-15 00:00:06,113 DEBUG [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
2012-04-15 00:00:06,113 DEBUG [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
2012-04-15 00:00:06,113 DEBUG [Old I/O client worker ([id: 0x53fd2626, /192.168.2.112:12051 => /83
```

**Potwierdzenie wysłania wiadomości do smsc operatora:**

```
2012-04-15 00:00:06,113 INFO [RMI TCP Connection(940)-192.168.2.75] [SEMAGateway] [20482876584] A
2012-04-15 00:00:06,113 INFO [RMI TCP Connection(940)-192.168.2.75] [SEMAGateway] [20482876584] M
```

**ORANGE****sms MO / IN / operator -> avantis****Przeczytanie surowej ramki z sieci (surowa w sensie - tablica bajtów albo string):**

```
2012-04-22 10:27:59,367 INFO [Old I/O client worker ([id: 0x7eb72e0a, /192.168.2.111:41356 => ucp
000/220412102759////3//50617373///0////////020100///18 on channel: [id: 0x7eb72e0a, /192.168.2.111:
```

**Przeparsowanie surowej ramki przychodzącej:**

```
2012-04-22 10:27:59,368 INFO [Old I/O client worker ([id: 0x7eb72e0a, /192.168.2.111:41356 => ucp
17, values={NMSG_AMSG_TMSG=[B@35e2e1a5, XSER={DCS_INFORMATION=[B@424c3346}, DCS=0, RPID=0000, MT=A
.2.111:41356 => ucpgw_new/194.9.223.143:3555]
2012-04-22 10:27:59,368 INFO [Old I/O client worker ([id: 0x7eb72e0a, /192.168.2.111:41356 => ucp
values={NMSG_AMSG_TMSG=[B@35e2e1a5, XSER={DCS_INFORMATION=[B@424c3346}, DCS=0, RPID=0000, MT=ALPHA
2012-04-22 10:27:59,368 INFO [Old I/O client worker ([id: 0x7eb72e0a, /192.168.2.111:41356 => ucp
2012-04-22 10:27:59,368 INFO [Old I/O client worker ([id: 0x7eb72e0a, /192.168.2.111:41356 => ucp
```

**Stworzenie payload objectu + Wysłanie wiadomości do MP:**

```
2012-04-22 10:27:59,378 INFO [Old I/O client worker ([id: 0x7eb72e0a, /192.168.2.111:41356 => ucp
555', sendDate=Sun Apr 22 10:27:59 CEST 2012, externalSendDate=Sun Apr 22 10:27:59 CEST 2012, deli
leTerminated=false, payloadDirection=IN, text='Pass', dataCodingScheme=0, partsNumber=1}
```

**Wysłanie ack'a do operatora:**

```
2012-04-22 10:27:59,382 INFO [Old I/O client worker ([id: 0x7eb72e0a, /192.168.2.111:41356 => ucp
]
```

**Wysłanie ramki (ACK):**

```
2012-04-22 10:27:59,382 INFO [Old I/O client worker ([id: 0x7eb72e0a, /192.168.2.111:41356 => ucp
values={}]] on channel: [id: 0x7eb72e0a, /192.168.2.111:41356 => ucpgw_new/194.9.223.143:3555]
```

**Wysłanie ramki (ACK) w postaci surowej:**

```
2012-04-22 10:27:59,382 INFO [Old I/O client worker ([id: 0x7eb72e0a, /192.168.2.111:41356 => ucp
b72e0a, /192.168.2.111:41356 => ucpgw_new/194.9.223.143:3555]
```

**sms MT / OUT / avantis -> operator****Odebranie wiadomości z MP:**

```
2012-04-22 10:28:19,425 INFO [RMI TCP Connection(155)-192.168.2.75] [UCPGateway] [20583302162] Me
externalSendDate=null, deliveryDate=null, validityDate=null, payloadType=TEXT_SMS, operator=POLAN
! Przydzielony Ci kod to M2773U28 . Dziękujemy za skorzystanie z naszego serwisu.', dataCodingSche
2012-04-22 10:28:19,425 INFO [RMI TCP Connection(155)-192.168.2.75] [UCPGateway] [20583302162] MP
e=null, deliveryDate=null, validityDate=null, payloadType=TEXT_SMS, operator=POLAND_ORANGE, master
i kod to M2773U28 . Dziękujemy za skorzystanie z naszego serwisu.', dataCodingScheme=0, partsNumbe
```

**Próba wysłania określonej ramki:**

```
2012-04-22 10:28:19,434 INFO [RMI TCP Connection(155)-192.168.2.75] [UCPUnorderedAcknowledgeManag
2012-04-22 10:28:19,434 INFO [RMI TCP Connection(155)-192.168.2.75] [UCPUnorderedAcknowledgeManag
2012-04-22 10:28:19,434 INFO [RMI TCP Connection(155)-192.168.2.75] [UCPUnorderedAcknowledgeManag
2012-04-22 10:28:19,434 INFO [RMI TCP Connection(155)-192.168.2.75] [UCPFrameEncoder] [2058330216
ORMATION=[B@7fc0c956}, MT=ALPHANUMERIC, OADC=92555, MCLS=1, ADC=603399376}] on channel: [id: 0x7eb
```

**Próba wysłania określonej ramki w postaci surowej:**

```
2012-04-22 10:28:19,434 INFO [RMI TCP Connection(155)-192.168.2.75] [UCPFrameEncoder] [2058330216
```

**Nadejście -> sparsowanie ramki ACK (z smsc operatora) + zmatchowanie:**



```

2012-04-22 10:28:19,509 INFO [Old I/O client worker ([id: 0x7eb72e0a, /192.168.2.111:41356 => ucp
No=33, values={ACK=A, SM=603399376:220412102819}] on channel: [id: 0x7eb72e0a, /192.168.2.111:4135
2012-04-22 10:28:19,509 INFO [Old I/O client worker ([id: 0x7eb72e0a, /192.168.2.111:41356 => ucp
2012-04-22 10:28:19,509 INFO [Old I/O client worker ([id: 0x7eb72e0a, /192.168.2.111:41356 => ucp
2012-04-22 10:28:19,509 INFO [Old I/O client worker ([id: 0x7eb72e0a, /192.168.2.111:41356 => ucp
51Ack[refNo=33, values={ACK=A, SM=603399376:220412102819}] to request: pl.avantis.agf.ucp.frames.F
55, MCLS=1, ADC=603399376}]

```

#### Potwierdzenie wysłania wiadomości do smsc operatora:

```

2012-04-22 10:28:19,509 INFO [RMI TCP Connection(155)-192.168.2.75] [UCPGateway] [20583302162] AL
2012-04-22 10:28:19,509 INFO [RMI TCP Connection(155)-192.168.2.75] [UCPGateway] [20583302162] MP

```

## PLUS

### sms MO / IN / operator -> avantis

#### Przeczytanie surowej ramki z sieci (surowa w sensie - tablica bajtów albo string):

```

2012-04-22 00:59:13,737 INFO [Old I/O client worker ([id: 0x64277469, 0.0.0.0/0.0.0.0:38313 => sm

```

#### Przeparsowanie surowej ramki przychodzącej:

```

2012-04-22 00:59:13,737 INFO [Old I/O client worker ([id: 0x64277469, 0.0.0.0/0.0.0.0:38313 => sm
2012-04-22 00:59:13,737 INFO [Old I/O client worker ([id: 0x64277469, 0.0.0.0/0.0.0.0:38313 => sm
2012-04-22 00:59:13,739 INFO [Old I/O client worker ([id: 0x64277469, 0.0.0.0/0.0.0.0:38313 => sm
2012-04-22 00:59:13,750 INFO [Old I/O client worker ([id: 0x64277469, 0.0.0.0/0.0.0.0:38313 => sm
2012-04-22 00:59:13,750 INFO [Old I/O client worker ([id: 0x64277469, 0.0.0.0/0.0.0.0:38313 => sm

```

#### Stworzenie payload obiektu + Wysłanie wiadomości do MP:

```

2012-04-22 00:59:13,760 INFO [Old I/O client worker ([id: 0x64277469, 0.0.0.0/0.0.0.0:38313 => sm

```

#### Wysłanie ack'a do operatora:

```

2012-04-22 00:59:13,766 INFO [Old I/O client worker ([id: 0x64277469, 0.0.0.0/0.0.0.0:38313 => sm

```

#### Wysłanie ramki (ACK):

```

2012-04-22 00:59:13,766 INFO [Old I/O client worker ([id: 0x64277469, 0.0.0.0/0.0.0.0:38313 => sm

```

#### Wysłanie ramki (ACK) w postaci surowej:

```

2012-04-22 00:59:13,766 INFO [Old I/O client worker ([id: 0x64277469, 0.0.0.0/0.0.0.0:38313 => sm

```

### sms MT / OUT / avantis -> operator

#### Odebranie wiadomości z MP:

```
2012-04-22 00:59:14,984 INFO [RMI TCP Connection(33)-192.168.2.75] [CIMDGateway] [20581165424] Me
2012-04-22 00:59:14,984 INFO [RMI TCP Connection(33)-192.168.2.75] [CIMDGateway] [20581165424] MP
```

### Próba wysłania określonej ramki:

```
2012-04-22 00:59:14,988 INFO [RMI TCP Connection(33)-192.168.2.75] [CIMDAcknowledgeManager] [2058]
2012-04-22 00:59:14,988 INFO [RMI TCP Connection(33)-192.168.2.75] [CIMDAcknowledgeManager] [2058]
2012-04-22 00:59:14,989 INFO [RMI TCP Connection(33)-192.168.2.75] [CIMDFrameEncoder] [2058]116542
```

**Próbeę wysłania określonej ramki w postaci surowej:**

```
2012-04-22 00:59:14,989 INFO [RMI TCP Connection(33)-192.168.2.75] [CIMDFrameEncoder] [2058116542]
```

**Nadejście -> sparsowanie ramki ACK (z smsc operatora) + zmatchowanie:**

```
2012-04-22 00:59:15,103 INFO [Old I/O client worker ([id: 0x64277469, 0.0.0.0/0.0.0.0:38313 => sm
2012-04-22 00:59:15,104 INFO [Old I/O client worker ([id: 0x64277469, 0.0.0.0/0.0.0.0:38313 => sm
2012-04-22 00:59:15,104 INFO [Old I/O client worker ([id: 0x64277469, 0.0.0.0/0.0.0.0:38313 => sm
```

**Potwierdzenie wysłania wiadomości do smsc operatora:**

```
2012-04-22 00:59:15,104 INFO [RMI TCP Connection(33)-192.168.2.75] [CIMDGateway] [20581165424] AI
2012-04-22 00:59:15,104 INFO [RMI TCP Connection(33)-192.168.2.75] [CIMDGateway] [20581165424] MP
```

## PLAY

**sms MO / IN / operator -> avantis**

### Przeczytanie surowej ramki z sieci (surowa w sensie - tablica bajtów albo string):

[illegible]

### Przeparasowanie surowej ramki przychodzącej:

```
2012-04-22 00:00:02,677 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
tatus=0, seq=13232, messageID=null, esmClass=0x0] on channel: [id: 0x0ca56214, 0.0.0.0/0.0.0.0:595
2012-04-22 00:00:02,677 DEBUG [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
=13232, messageID=null, esmClass=0x0]
2012-04-22 00:00:02,677 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:02,677 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:02,677 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:02,677 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:02,677 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:02,677 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:02,677 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:02,677 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:02,677 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:02,677 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
on bits 3 and 2: 0
2012-04-22 00:00:02,677 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
me, value: 0
2012-04-22 00:00:02,677 DEBUG [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
gle part
2012-04-22 00:00:02,677 DEBUG [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:02,680 WARN [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:02,680 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:02,680 DEBUG [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:02,693 DEBUG [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
heService): totalTime=414794, totalCount=37427
2012-04-22 00:00:02,693 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
se
```

**Wysłanie wiadomości do MP:**

```
2012-04-22 00:00:02,751 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
```

**Zapisanie w cache'u:**

```
2012-04-22 00:00:02,752 DEBUG [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:02,762 DEBUG [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:02,763 DEBUG [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
```

**Wysłanie ack'a do operatora:**

```
2012-04-22 00:00:02,763 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
```

**Wysłanie ramki (ACK):**

```
2012-04-22 00:00:02,763 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
```

**Wysłanie ramki (ACK) w postaci surowej:**

```
2012-04-22 00:00:02,763 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
```

**sms MT / OUT / avantis -> operator****Odebranie wiadomości z MP:**

```

2012-04-22 00:00:05,107 INFO [RMI TCP Connection(85)-192.168.2.75] [SMPPGateway] [20580920309] Me
2012-04-22 00:00:05,107 DEBUG [RMI TCP Connection(85)-192.168.2.75] [SMPPGateway] [20580920309] Ch
2012-04-22 00:00:05,108 INFO [RMI TCP Connection(85)-192.168.2.75] [SMPPGateway] [20580920309] MP

```

**Przekodowanie ramki i ustawienie kodowania:**

```

2012-04-22 00:00:05,108 INFO [RMI TCP Connection(85)-192.168.2.75] [SmsPduConverter] [20580920309]
2012-04-22 00:00:05,108 INFO [RMI TCP Connection(85)-192.168.2.75] [SmsPduConverter] [20580920309]

```

**Podzielenie wiadomości i wrzucenie jej do cache'a:**

```

2012-04-22 00:00:05,108 DEBUG [RMI TCP Connection(85)-192.168.2.75] [SMPPGateway] [20580920309] SM
2012-04-22 00:00:05,108 DEBUG [RMI TCP Connection(85)-192.168.2.75] [RMICConnector] [20580920309] i
2012-04-22 00:00:05,112 DEBUG [RMI TCP Connection(85)-192.168.2.75] [RMIClientInterceptorProxy] [2
2012-04-22 00:00:05,112 DEBUG [RMI TCP Connection(85)-192.168.2.75] [SMPPWindowingChannelPool] [20
2012-04-22 00:00:05,112 DEBUG [RMI TCP Connection(85)-192.168.2.75] [SMPPAcknowledgeManager] [2058

```

**Próba wysłania określonej ramki:**

```

2012-04-22 00:00:05,112 INFO [RMI TCP Connection(85)-192.168.2.75] [SMPPFrameEncoder] [20580920309]

```

**Próbę wysłania określonej ramki w postaci surowej:**

```

2012-04-22 00:00:05,112 INFO [RMI TCP Connection(85)-192.168.2.75] [SMPPFrameEncoder] [20580920309]

```

**Nadejście -> sparsowanie ramki ACK (z smsc operatora):**

```

2012-04-22 00:00:05,180 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:05,180 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:05,180 INFO [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:05,181 DEBUG [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:05,181 DEBUG [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:05,181 DEBUG [Old I/O client worker ([id: 0x0ca56214, 0.0.0.0/0.0.0.0:59562 => /8
2012-04-22 00:00:05,181 DEBUG [RMI TCP Connection(85)-192.168.2.75] [SMPPGateway] [20580920309] Su

```

**Potwierdzenie wysłania wiadomości do smsc operatora:**

```

2012-04-22 00:00:05,181 INFO [RMI TCP Connection(85)-192.168.2.75] [SMPPGateway] [20580920309] Al
2012-04-22 00:00:05,181 INFO [RMI TCP Connection(85)-192.168.2.75] [SMPPGateway] [20580920309] MP

```

**Sposoby badania ruchu na bramkach**

W celu sprawdzenia ruchu IN/OUT na danej usłudze/bramce można zastosować 2 metody.

**Sprawdzanie na bazie**

1. Sprawdzanie w mp.mpevent w odpowiedni sposób pól SENT\_MESSAGES\_NUM i DELIVERED\_MESSAGES\_NUM. Do mp.mpevent wliczane są również potwierdzenia ( USER ACK, SMSC ACK) dlatego należy o tym pamiętać badając ruch na bramkach za pomocą odpowiednich selectów na tabeli mp.mpevent.

Poniższe zapytanie zwraca liczbę wiadomości OUT/IN na danej usłudze w wybranym czasie ( creation\_time):

```
SELECT service_id, SUM(delivered_messages_num) AS RUCH_OUT, SUM(sent_messages_num) AS RUCH_IN
FROM mp.mpevent e
WHERE service_id = '3'
AND creation_time >= to_date('10-11-2010 00:00:00', 'dd-mm-yyyy hh24:mi:ss')
AND creation_time <= to_date('10-11-2010 01:00:00', 'dd-mm-yyyy hh24:mi:ss')
GROUP BY service_id
ORDER BY service_id
```

Podany poniżej select zwróci statystyki ruchu IN/OUT dla bramki tmobile-sk-gateway-bo( mpid: 8208) z dnia 2010-11-10. Przydatny w przypadku gdy trzeba wyskalować reguły ruchowe dla danej usługi.

```
SELECT service_id, ROUND(creation_time, 'hh24'), SUM(delivered_messages_num) AS RUCH_OUT, SUM(sent_messages_num) AS RUCH_IN
FROM mp.mpevent e
WHERE service_id = '8208'
AND creation_time >= to_date('2010-11-10', 'yyyy-mm-dd')
AND creation_time <= to_date('2010-11-11', 'yyyy-mm-dd')
GROUP BY service_id, ROUND(creation_time, 'hh24')
ORDER BY service_id, ROUND(creation_time, 'hh24')
```

## Sprawdzanie w logach bramek za pomocą przykładowego tail/cat/grep'a.

**SMS MO (Mobile Originated)**- wiadomość z telefonu użytkownika do sieci komórkowej.

**SMS MT (Mobile Terminated)**- wiadomość z sieci komórkowej do telefonu użytkownika.

## ERA / PLUS / ORANGE / PLAY / CYFROWY POLSAT/ SILVERSTREET / CARDBOARDFISH

W zależności od operatora podmieniamy nazwę logu bramki

### RUCH OUT(MT):

Liczba wysłanych wiadomości do SMSC od rozpoczęcia ostatniej godziny.

```
cat /usr/local/logs/mpservices/comm/era-sms-gateway.log | grep "`date +%y-%m-%d\ %H:%M`" | sed s/..
```

Liczba wysłanych wiadomości do SMSC od rozpoczęcia ostatnich 10 min.

```
tail -100000 /usr/local/logs/mpservices/comm/era-sms-gateway.log | grep "`date +%y-%m-%d\ %H:%M`" |
```

Liczba wysłanych wiadomości do SMSC w trakcie trwania ostatniej minuty.

```
tail -10000 /usr/local/logs/mpservices/comm/era-sms-gateway.log | grep "`date +%y-%m-%d\ %H:%M`"
```



**RUCH IN(MO):**

Liczba odebranych wiadomości od SMSC od rozpoczęcia ostatniej godziny.

```
cat /usr/local/logs/mpservices/comm/era-sms-gateway.log | grep "`date +%y-%m-%d\ %H:%M | sed s/..
```

Liczba odebranych wiadomości z SMSC od rozpoczęcia ostatnich 10 min.

```
tail -50000 /usr/local/logs/mpservices/comm/era-sms-gateway.log | grep "`date +%y-%m-%d\ %H:%M |
```

Liczba odebranych wiadomości do SMSC w trakcie trwania ostatniej minuty.

```
tail -10000 /usr/local/logs/mpservices/comm/era-sms-gateway.log | grep "`date +%y-%m-%d\ %H:%M`"
```

**Czas ważności wiadomości**

W wiadomościach wysyłanych do poszczególnych operatorów istnieje taki atrybut validityDate, oznacza on datę wygaśnięcia ważności wiadomości, co w praktyce oznacza, że gdy ta data zostanie przekroczona wiadomość nie będzie ponownie dostarczana do użytkownika tylko pójdzie w zapomnienie. Ten parametr jest ważny tylko w przypadku niedostępności użytkownika końcowego (np. wyłączony telefon). Każdy z operatorów ma właściwy dla siebie maksymalny czas validityDate, dla każdej wiadomości standardowo jest ustawiana właśnie wartość maksymalna (wyjątkiem jest PowerBulkman)

1. ERA - 72 godziny
2. PLAY - 72 godziny
3. ORANGE - 6 godzin
4. PLUS
  - default 24 godziny - max 7\*24godziny

**Monitor Pool WatchDog**

Podczas problemów z siecią zdarza się, że niektóre kanały nie próbują nawiązać ponownie połączenia z SMSC, lub ChannelIdleStateManager nie wywołuje na nich wiadomości keepAlive.

Zaimplementowany mechanizm monitoruje oba przypadki :

Ruch na kanale:

- Każda wiadomość przechodząca przez nettiego aktualizuje timestamp kanału.
- WatchDog sprawdza czy dla każdego kanału ostatnia aktywność nie przekroczyła czasu timeouta
- Zrzuca MPEvent FATAL - No activity on channel
- Wymusza zamknięcie kanału i ponowne jego podłączenie.

Podłączenie kanału:

- Nadzorca uruchamia się z częstotliwością = alivePeriod + aliveResponseTimeout
- Sprawdza czy ilość podłączonych kanałów jest równa poolSize
- Jeżeli nie - sprawdza czy jakieś kanały są w trakcie nawiązywania połączenia
- Jeżeli nie - zrzuca MPEvent FATAL - No connection on gateway i wymusza podłączenie brakujących kanałów.

**Logi:**

Monitorowanie ruchu podłączonych kanałów:

```
2016-04-27 10:47:35,482 INFO [watch-dog-executor-0] [CIMDWindowingChannelPool] [] Checking traffi
68.168.52:7897]=1461746835457, [id: 0x4b42c568, 0.0.0.0/0.0.0.0:36106 => /192.168.168.52:7897]=146
168.168.52:7897]=1461746835458}
2016-04-27 10:47:35,482 INFO [watch-dog-executor-0] [CIMDWindowingChannelPool] [] Channel with id
2016-04-27 10:47:35,482 INFO [watch-dog-executor-0] [CIMDWindowingChannelPool] [] Channel with id
2016-04-27 10:47:35,483 INFO [watch-dog-executor-0] [CIMDWindowingChannelPool] [] Channel with id
2016-04-27 10:47:35,483 INFO [watch-dog-executor-0] [CIMDWindowingChannelPool] [] Checking monito
```

Monitorowanie ilości podłączonych kanałów:

```
2016-04-27 10:52:35,507 INFO [watch-dog-executor-0] [CIMDWindowingChannelPool] [] Checking monito
2016-04-27 10:52:35,507 WARN [watch-dog-executor-0] [CIMDWindowingChannelPool] [] Some channels a
```

## MONITORING

### AGF Gateways errors

Monitoring szuka wystąpień konkretnych błędów na wszystkich bramkach AGF.

#### Szukane błędy:

```
'%Timed out for message%' - Eventy rzucane jak operator nie wyśle acka albo my go nie odbierzemy
'%Missing ack processed%' - Eventy rzucane jak ack zgubi się na bramce i awaryjnie zostanie przep
'%No connection on gateway%' - Eventy rzucane jak bramka nie może się podnieść po zerwaniu połącze
'%No activity on channel%' - Eventy rzucane jak connection Keep Alive nie działa
'%Some channels probably not connected%' - Eventy rzucane jak niektóre kanały nie mogą połączyć si
```

#### Częstotliwość:

Aktualnie monitoring alertuje, gdy wykryje powyżej 5 błędów w przeciągu 15 min.

#### Weryfikacja:

**Błędów per bramka**

```
select count(*), e.service_id as sid, s.name
from mp.mpservice s, mp.mpevent e where
e.service_id=s.service_id and e.mpevent_level >= 1
and (s.name like '%-sms-gateway%'
     or s.name like '%gateway-agf%'
     or s.name like '%agf-gateway%'
     or s.name like 'plus-sms%'
     or s.name like 'ucp-gateway%')
and (e.mpevent_desc like '%Timed out for message%'
     or e.mpevent_desc like '%Missing ack processed%'
     or e.mpevent_desc like '%No connection on gateway%'
     or e.mpevent_desc like '%No activity on channel%'
     or e.mpevent_desc like '%No more new connections allowed for%'
     or e.mpevent_desc like '%Some channels probably not connected%')
and e.creation_time >= sysdate-.25/24
group by e.service_id, s.name
having count(*) > 5;
```

**Lista błędów**

```
select e.creation_time as time, e.service_id, s.name, e.mpevent_level as "level", e.mpevent_desc as desc
from mp.mpservice s, mp.mpevent e where
e.service_id=s.service_id and e.mpevent_level >= 1
and (s.name like '%-sms-gateway%'
     or s.name like '%gateway-agf%'
     or s.name like '%agf-gateway%'
     or s.name like 'plus-sms%'
     or s.name like 'ucp-gateway%')
and (e.mpevent_desc like '%Timed out for message%'
     or e.mpevent_desc like '%Missing ack processed%'
     or e.mpevent_desc like '%No connection on gateway%'
     or e.mpevent_desc like '%No activity on channel%'
     or e.mpevent_desc like '%No more new connections allowed for%'
     or e.mpevent_desc like '%Some channels probably not connected%')
and e.creation_time >= sysdate-.35/24
order by e.creation_time desc;
```

**Reakcja:**

- Weryfikacja błędów - Należy zweryfikować czy problem występuje dla 1 bramki czy może problem jest dla całego operatora
- W przypadku 3 ostatnich eventów spróbować restartu bramki
- Jeżeli problem jest po stronie operatora, należy wysłać do niego zgłoszenie.

Pomocnym w określeniu awarii może być developer i Katarzyna Osytek.

**PLAY-MT BLEDY**

Monitoring szuka w mpeventach czy błąd "Response was of unexpected" w ostatnich 15 minutach wystąpił więcej niż 2 razy.

**Weryfikacja:**

**Lista błędów**

```
select e.creation_time as time, e.queue_size as queue, e.delivered_messages_num as delivered,  
e.sent_messages_num as sent, e.service_id, s.name, e.mpevent_level as "level", e.mpevent_desc as d  
from mp.mpsservice s, mp.mpevent e where  
e.service_id=s.service_id and  
e.service_id = 100609  
and e.mpevent_desc like '%Response was of unexpected%'  
and e.creation_time >= sysdate-0.5/24  
order by e.creation_time desc;
```

**Reakcja:**

- weryfikacja logów bramki play-mt-gateway
- w przypadku błędu po stronie operatora, wysłanie zgłoszenia

👍 Like Be the first to like this