

Zastosowanie otwartych technologii informatycznych do budowy aplikacyjnej infrastruktury Smart Grids

Streszczenie. Rozwój aplikacji informatycznych w obszarze sieci inteligentnych stanowi jeden z podstawowych czynników warunkujących uzyskanie wysokiej, rynkowej efektywności Smart Grids. Aplikacjom tym stawiane są wymagania gwarantujące niezawodność i pewność działania, zachowując jednocześnie nieograniczone bezpieczeństwo gromadzonych danych. W artykule przedstawiono ocenę możliwości zastosowania w sieciach inteligentnych otwartych technologii informatycznych wraz z propozycją modelu takiej aplikacji.

Abstract. Development of computer applications in the field of intelligent networks is one of the key factors for getting high, the market efficiency of Smart Grids. Applications that are placed on the requirements to ensure reliable operation, while maintaining safety unlimited data collected. This paper presents an assessment of the applicability of the intelligent network of open technologies, including proposed model of the application.. (**The use of open technologies for building application of the Smart Grids**).

Słowa kluczowe: sieci inteligentne, otwarte systemy informatyczne, architektura klastrowa.

Keywords: smart grids, open source, cluster architecture.

doi:10.12915/pe.2014.03.14

1. Wstęp

Postępująca restrukturyzacja przedsiębiorstw energetycznych polegające na demonopolizacji, prywatyzacji, konsolidacji w grupy energetyczne, *unbundling* na spółki dystrybucyjne i handlowe, czy też zmiany prawa energetycznego wprowadzające nowe byty w systemie elektroenergetycznym, takie jak prosumenci, wprowadzają konieczność ciągłego dostosowywania infrastruktury i systemów IT w elektroenergetyce do potrzeb i wymagań przeobrażającego się rynku. Aby system energetyczny działał przy jednoczesnym udziale tak wielu podmiotów wytwórczych, dystrybucyjnych oraz handlowych niezbędne jest dokładne opomiarowanie sieci na wszystkich jej poziomach. Konieczne staje się zbilansowanie całego systemu elektroenergetycznego. Wychodząc naprzeciw tym potrzebom powstały różne koncepcje zdalnego opomiarowania sieci: począwszy od jednokierunkowego odczytu liczników – AMR (ang. *Automatic Meter Reading*) wykonywanego z określonym interwałem czasu, poprzez koncepcje komunikacji dwukierunkowej z ograniczonymi komendami sterowania – AMM (ang. *Advanced Metering Management*), ewoluując w kierunku zdalnej, ciągłej, dwukierunkowej komunikacji i wymiany danych z licznikami – AMI (ang. *Advanced Metering Infrastructure*) [6, 7, 8]. Samo gromadzenie i archiwizowanie danych to tylko część działań operacyjnych, wymaganych przez rynek energii elektrycznej. Aby w pełni sprostać wymaganiom nowego kierunku przekształceń w elektroenergetyce niezbędne są systemy informatyczne umożliwiające gromadzenie, przetwarzanie i zarządzanie danymi – MDM (ang. *Meter Data Management*). Połączenie systemów AMI oraz MDM daje podstawy do użycia pojęcia inteligentnych sieci energetycznych (ang. *Smart Grids*). W ramach inteligentnych sieci wiele uwagi poświęca się interakcjom systemu Smart Grids z wszystkimi uczestnikami rynku. Rozwija się także proces zaangażowania Klientów w bilansowanie rynku poprzez racjonalność wykorzystania energii elektrycznej.

2. Wymagania stawiane systemom IT w obszarze Smart Grids

Dynamiczny rozwój rynku energii, a także przemiany na nim zachodzące stawiają przed systemami informatycznymi wysokie wymagania. Systemy te podlegają stałym modyfikacjom funkcjonalnym. Istotnym uwarunkowaniem rozwoju systemów informatycznych jest postawienie na ich

otwartość. Systemom informatycznym w obszarze Smart Grids stawiane jest jedno zasadnicze wymaganie gwarantujące niezawodności działania. Systemy te powinny być tworzone w myśl zasady *No Single Point of Failure* (z ang. bez pojedynczego punktu awarii). Niemal idealnym rozwiązaniem byłoby ustanowienie minimalnego współczynnika utrzymania dostępności systemu – SLA (ang. *Service Level Agreement*) dla systemu Smart Grids na poziomie 99,99%, co przekłada się na możliwość braku dostępności usług oferowanych przez system przez niespełna 53 minuty w ciągu roku.

Zastosowanie otwartego oprogramowania do budowy systemów IT powinno przede wszystkim zapewniać skalowalność, dużą dostępność i niewielką cenę implementacji takich systemów w stosunku do systemów komercyjnych a także gwarantować bezpieczne przechowywanie pozyskiwanych danych. Systemy IT w obszarze Smart Grids mają za zadanie gromadzić i przetwarzać oprócz danych osobowych klientów, również informacje o profilu zużycia, sytuacji technicznej sieci oraz dodatkowo zapewniać możliwość kontroli nad systemem SCADA (ang. *Supervisory Control And Data Acquisition*) poprzez wysyłanie komend sterujących systemem. Dostęp osób niepowołanych do tego rodzaju systemu mógłby mieć więc katastrofalne skutki związane nie tylko z kradzieżą danych osobowych, ale również z załamaniem systemu energetycznego poprzez przejście kontroli nad systemami sterowania. Systemy IT w obszarze Smart Grids powinny mieć rozbudowane mechanizmy szyfrowania transmisji danych oraz zaawansowane mechanizmy autoryzacji i autentykacji oraz zarządzania ich uprawnieniami. Mówiąc o bezpieczeństwie systemu, nie można myśleć tylko i wyłącznie o odporności na ataki celowe, lub błędy użytkowników. Bezpieczeństwo powinno być również zapewnione poprzez stosowanie mechanizmów replikacji danych (ang. *data replication*), mechanizmów wykonywania kopii zapasowych (ang. *backup copy*), automatycznych procedur naprawczych oraz mechanizmów logowania błędów i ich raportowania do osób odpowiedzialnych za utrzymanie systemu.

Wymaganiem na równi istotnym z niezawodnością jest szeroko pojęta szybkość rozwoju oprogramowania. Pod pojęciem tym kryje się poprawa wykrytych błędów, jak również aktualizowanie, oraz poszerzanie funkcjonalności, nadążając za aktualnymi potrzebami. System informatyczny Smart Grids powinien być gotowy na ciągłą rozbudowę.

Otwartość systemów w tym aspekcie również jest nieoceniona. Sprawia ona, wielu firmom lub programistom indywidualnym możliwość promowania swoich pomysłów w rozwoju systemów. Fakt ten jest bardzo istotny i ważny podkreślenia, ponieważ takie podejście do rozwoju oprogramowania uniezależnia podmioty z niego korzystające od konkretnego dostawcy usług IT. Uniezależnienie od konkretystania z usług IT przez podmioty korzystające z otwartego oprogramowania powoduje zmniejszenie kosztów utrzymania i rozwoju oprogramowania. Wprowadza to konkurencyjność usług IT, a przez to i spadek cen usług (np. aktualizacji oprogramowania) w stosunku do wykorzystywania oprogramowania komercyjnego, którego rozwój zapewnia zazwyczaj tylko i wyłącznie jego wytwórca.

W przypadku systemów Smart Grids, które są systemami rozproszonymi jednymi z kluczowych wymagań są skalowalność horyzontalna i wysoka wydajność. Skalowalność systemów komputerowych rozumie się poprzez zdolność systemu do zachowania parametrów jakościowych (wydajności), w przypadku zwiększenia liczby przetwarzanych danych uzyskiwana poprzez rozbudowę infrastruktury, w oparciu o którą działa system. Jako skalowalność horyzontalną rozumie się zwiększanie wydajności systemu poprzez zwiększanie liczby serwerów. Zgodnie z założeniami w Polsce ma zostać zainstalowanych ok. 15 milionów inteligentnych liczników do 2020 roku [1,2], niezbędna będzie również rozbudowa systemów IT obsługujących te liczniki. Szacując, że dane bilingowe wraz z danymi technicznymi pochodzące z jednorazowego odczytu licznika energii elektrycznej zajmą ok. 1 KB danych, to już zebrane z 15 mln liczników zajmą ok. 15 GB danych. Przy tak dużej jednorazowej partii danych niezbędne jest zbudowanie odpowiednich łączy do ich przesłania, ale także zbudowanie systemu IT – skalowalnego i wydajnego, który będzie w stanie te dane zapisywać w bazie danych i przetworzyć w jednostce czasu. Biorąc pod uwagę stanowisko Prezesa URE, które zakłada sczytywanie danych z licznika co 15 minut [3], należy zwrócić uwagę na zwiększającą się niezmiernie liczbę danych do przesyłania, zapisania i przetworzenia. W liczbach przedstawia się to następująco: 1 KB danych z jednorazowego odczytu licznika zwiększy się w ciągu doby do 96 KB danych, co w skali 15 milionów liczników daje 1,44 TB danych do przetworzenia w ciągu doby. Systemy bazodanowe gromadzące tak duże ilości danych w ciągu doby powinny działać ze względów bezpieczeństwa jak i wydajności w architekturze klastrowej, czyli systemów wzajemnie powiązanych. Aby system działał wydajnie powinien posiadać mechanizmy kolejkowania danych (ang. *data queue*) do przetwarzania oraz mechanizmy równoważenia obciążenia obliczeniowego (ang. *load balancing*) dla realizowanych procesów. Jak już wspomniano systemy Smart Grids oprócz gromadzenia danych mają za zadanie ich przetwarzanie, udostępnianie i raportowanie stanu systemu energetycznego. Aby przetworzyć w bardzo krótkim czasie olbrzymie ilości danych niezbędne jest zastosowanie sprawnych mechanizmów ich analizy. Należy tu zwrócić uwagę na stosowanie mechanizmów „NoSQL” (ang. *Not only SQL*), które umożliwią szybkie przetworzenie zebranych, poetykietowanych danych pod kątem uzyskania określonych danych wynikowych (np. aktualnych raportów na temat działania sieci). Przy wymaganiach systemu IT dla Smart Grids warto również wspomnieć o wybraniu wydajnego, rozproszonego ze względu na bardzo duże ilości danych które będzie on gromadził oraz odpornego na awarie systemu plików. Systemy IT dla Smart Grids powinny mieć możliwość realizacji bardzo dużej

jednoczesnej obsługi żądań, co wymaga wysoko dostępnych usług hostujących aplikacje Smart Grids, które będą działać zarówno w schemacie asynchronicznym jak i synchronicznym. Dane zgromadzone przez System Smart Grids będą zapewne służyć wielorakim analizom. Aby analizować pod różnym kątem tak duże ilości zgromadzonych danych system IT dla Smart Grids powinien być zaopatrzony w mechanizmy hurtowni danych (ang. *data warehouse*).

Od systemów IT dla Smart Grids wymagana jest wysoka interoperacyjność z innymi systemami dziedzinowymi. System taki powinien posiadać możliwość dwukierunkowej wymiany danych poprzez standardowe interfejsy komunikacyjne (otwarte protokoły sieciowe) oraz otwarte interfejsy wymiany danych (np. *WebServices* opisane przez WSDL (ang. *Web Services Description Language*) wykorzystujące SOAP 1.2 (ang. *Simple Object Access Protocol*), XML (ang. *Extensible Markup Language*), JSON (ang. *JavaScript Object Notation*)). Oprócz tego, że interfejsy komunikacyjne będą odpowiedzialne między innymi za dwukierunkową komunikację z koncentratorami, do których z kolei podłączone będą inteligentne liczniki, ale również np. za przekazywanie danych pomiarowych do systemów informatycznych sprzedawców energii, na podstawie których wystawione zostaną faktury, czy też będą one wykorzystane np. jako połączenie z bramą do inteligentnego budynku (HAN).

3. Model architektury systemu IT w sferze Smart Grids zbudowany w oparciu o otwarte technologie informatyczne

Idee propagowania oprogramowania wolnego (ang. *free software*) oraz otwartego (ang. *open source*) towarzyszą rozwojowi informatyki od samego początku. Oprócz organizacji informatycznych idee te szeroko propaguje również Unia Europejska. Wolne i otwarte oprogramowanie daje użytkownikom prawa do swobodnego uruchamiania, kopiowania, rozpowszechniania, analizowania, zmian i ulepszania programów. Licencje wolnego i otwartego oprogramowania zostały sformalizowane przez organizacje *Free Software Foundation* (z ang. Fundacja Wolnego Oprogramowania) oraz *Open Source Initiative*. Istotną cechą wolnego i otwartego oprogramowania jest to, że korzystanie z niego jest bezpłatne. Zmniejszają się więc koszty zakupu i utrzymania systemów IT zbudowanych w oparciu o wolne i otwarte oprogramowanie. Z wolnego i otwartego oprogramowania korzystają tacy globalni potentaci w na rynku IT jak chociażby Amazon, Google, Flickr, Facebook, Twitter czy Instagram. Wiele państw, jak chociażby Islandia, zdecydowało się również na używanie wolnego i otwartego oprogramowania w swoich jednostkach państwowych. Odpowiednio dobrane wolne i otwarte oprogramowanie niczym nie ustępuje oprogramowaniu komercyjnemu. Dla przykładu serwis Pingdom poddał w 2012 roku badaniu 10 tysięcy najpopularniejszych stron internetowych pod względem miejsca ich składowania. Okazuje się, że niemal 75 % jest umieszczona na serwerach Open Source.

Tradycyjny model użytkowania systemów IT zakładał, że system jest ściśle powiązany ze sprzętem, na którym został zainstalowany i jest on niemalże nieprzenaszalny na inne platformy sprzętowe. Stosowanie takiego modelu powoduje wiele problemów w przypadku awarii sprzętu, bądź też samego systemu. Stwarza nierzadko konieczność czasochłonnej instalacji systemu. Model ten hamuje również rozwój systemu w przypadku chęci jego używania na bardziej wydajnym sprzęcie. Chęć przeniesienia systemu do bardziej wydajnej infrastruktury IT wiąże się z ponowną instalacją systemu oraz z migracją danych z

poprzedniej instalacji systemowej. Podnosi to koszty serwisowe systemu, oraz zwiększa ryzyko braku dostępności do systemu. Sposobem na poradzenie sobie ze wspomnianymi problemami jest wirtualizacja. Wirtualizacja (ang. *Virtualization*) – jest to szerokie pojęcie odnoszące się do rozwiązań programowych i/lub sprzętowych umożliwiających oddzielenie wykorzystywanych zasobów sprzętowych od systemów i/lub aplikacji, które z tych zasobów korzystają. W środowisku zwirtualizowanym systemy IT poprzez działanie na maszynach wirtualnych są oddzielone od infrastruktury fizycznej (sprzętowej) przez tzw. platformę wirtualizacyjną. Sprawia to, że stają się one niezależnym bytem, który jest łatwo przenaszalnym pomiędzy infrastrukturą fizyczną, z której korzystają systemy. Nierzadko dzięki wirtualizacji migracja systemów IT pomiędzy fizycznymi maszynami odbywa się „w locie” tzn. bez konieczności wyłączenia systemów. Działający system zostaje przeniesiony na inną maszynę fizyczną w stanie niezmiennym. Operacja przeniesienia jest niezauważalna dla użytkowników korzystających aktywnie z systemu. Oprócz przenaszalności bardzo przydatną cechą gwarantującą wysokie bezpieczeństwo systemów oraz podnoszącą poziom dostępności aplikacji jest możliwość zapisywania i odtwarzania stanów systemów w danej chwili (ang. *snapshot*). Popularną platformą wirtualizacji jest pręźnie rozwijany system KVM wywodzący się wprost z jądra systemu operacyjnego Linux. KVM działa w trybie wirtualizacji wspieranej sprzętowo, w której wirtualizowany system operacyjny działa niczym na prawdziwym, fizycznym sprzęcie. Odwołania systemu wirtualnego do tych elementów fizycznych sprzętu, które mogłyby mieć wpływ na działanie innych systemów wirtualnych znajdujących się na tym samym sprzęcie, są przechwytywane przez środowisko wirtualizacyjne, a następnie emulowane,

Systemy operacyjne są zasadniczą i nieodzowną częścią wszelkich systemów komputerowych [4]. Wybierając otwarty system operacyjny, czyli środowisko służące do zarządzania infrastrukturą sprzętową oraz środowisko uruchomieniowe dla aplikacji swoją uwagę warto skupić na systemie Linux. Pod pojęciem Linux kryje się bardzo wiele. Linux jest wolnym oprogramowaniem wywodzącym się z rodziny uniksowych systemów operacyjnych, którego rozwój rozpoczął się w 1991 r. Wolność i otwartość Linuxa, a zarazem jego rozwój jest związany z pełnym dostępem do jego kodu źródłowego. Linux – formalnie oznacza jądro systemu. Dystrybucja Linux to już z kolei kompletny system operacyjny składający się z jądra Linux, bibliotek systemowych, oraz pomocy systemowych (ang. *system utilities*). System ten jest dystrybuowany na zasadzie licencji GNU (z ang. *General Public License*), dlatego też każdy, kto modyfikuje jego kod źródłowy jest zobowiązany udostępnić go innym. Stąd w ramach różnych projektów skupiających wokół siebie grupy programistów następuje szybki rozwój produktu. Stawiając na niezawodność infrastruktury IT dla aplikacji Smart Grids warto skupić się na dystrybucji Linux Debian. Dystrybucja ta realizowana przez ochotników z całego świata cechuje się bardzo dużą stabilnością przez wzgląd na korzystanie jedynie ze stabilnych (sprawdzonych) wersji jądra systemu. Stabilne jądro posiadają minimalną ilość błędów, a odkryte błędy natychmiast są poprawiane, co podnosi bezpieczeństwo. Debian może działać przez bardzo długi okres czasu bez konieczności restartu. Dystrybucję tę charakteryzuje również znakomite wsparcie społeczności. Debiana używa wiele organizacji i indywidualnych osób. Wersje tej dystrybucji pojawiają się dość rzadko w stosunku do innych dystrybucji. Za to są bardzo dobrze sprawdzone nim wejść do użytku. Ta wersja systemu posiada minimalne

wymagania sprzętowe poprzez niewielką użycie pamięci operacyjnej. System ten skupia się przede wszystkim na funkcjonalnościach, bez zbędnych elementów GUI, które mogłyby wpływać na jego słabszą wydajność. Dystrybucja ta, jak i inne charakteryzuje się wielodostępnością, co oznacza, że na jednej maszynie może pracować wielu zalogowanych użytkowników jednocześnie. Debiana charakteryzuje wysoka konfigurowalność systemu i swoboda w instalowaniu zawsze aktualnych rozszerzeń – pakietów systemowych. Nie bez znaczenia jest również łatwość aktualizacji systemu. Właściwie każdy aspekt systemu można dostosować do własnych potrzeb. Wydajność Debiana jako systemu wielowątkowego, we wszelkich rankingach wydajności stawia go w ścisłej czołówce. Dystrybucja ta charakteryzuje się znakomitym zarządzaniem urządzeniami sieciowymi, co w przypadku usieciowienia aplikacji na potrzeby Smart Grids będzie miało duże znaczenie.

W przypadku systemów dla Smart Grids, istotnym problemem jest przechowywanie olbrzymiej ilości danych umożliwiające szybkie wyszukiwanie oraz edycję. Dokonując wyboru systemu zarządzania bazami danych (SZBD, ang. *Databases Management System - DBMS*) w przypadku Smart Grids warto swoją uwagę skierować na system bazodanowy MySQL. Cechuje się on potencjałem największych możliwości pod kątem przechowywania ilości danych, wydajności, możliwości konfiguracji oraz największą popularnością na świecie wśród systemów bazodanowych o otwartym kodzie. Oprócz standardowych funkcjonalności baz danych takich jak relacyjny model danych z kluczami głównymi i obcymi, indeksami, procedurami wbudowanymi, własnymi typami danych, etc., którymi mogą się pochwalić wszystkie wspomniane produkty bazodanowe, istotną cechą bazy MySQL jest możliwość replikacji danych w strukturze klastrowej (ang. *Cluster Architecture*), czyli powiązanych ze sobą systemów rozproszonych. Mechanizm ten ma na celu aktualizację wielu instancji tej samej bazy danych znajdującej się na różnych maszynach, przez co system bazodanowy staje się bardziej wydajnym i niezawodnym przez działanie w schemacie rozproszonym. Domyślna replikacja w klastrze MySQL polega na zapisywaniu zmian danych, które zachodzą w instancji bazy pełniącej rolę Master oraz asynchronicznym przepisaniu tych samych zmian do klastrowo połączonych instancji bazy danych pełniących rolę Slave. Ten mechanizm replikacji (Master-Slave) nie narzuca żadnego okresu maksymalnego ani minimalnego, po którym musi dojść do zreplikowania danych pomiędzy węzłami. Może więc powodować pewne problemy przy ewentualnych awariach sprzętowych głównie maszyny, na której znajduje się instancja Master, w sytuacji, gdzie migawka zawierająca zmienione dane nie została jeszcze zreplikowana na instancji Slave. Zdecydowanie bezpieczniejszy byłby mechanizm replikacji Multi-Master (ang. *multimaster replication*). W tym schemacie replikacji wszystkie węzły są równoważne, co oznacza, że każdy z węzłów jest węzłem nadrzędnym. Zmiany wprowadzone w którymkolwiek węzle zostają zreplikowane we wszystkich pozostałych węzłach środowiska klastrowego. Obiekty są więc identyczne we wszystkich węzłach w każdej chwili. Otwartość kodu MySQL daje możliwość rozwoju własnych implementacji bazy danych z rozszerzoną funkcjonalnością pisanie tzw. *pluginów* (z ang. *wtyczek*). Ciekawymi wtyczkami do MySQL są Galera Cluster for MySQL oraz MySQL *Write Set Replication* (MySQL-wsrep) API. Są to projekty działające na zasadach *open source*, które cieszą się dużą popularnością w rozwiązaniach. Zasadnicze wymogi, jakie są im stawiane to stabilność i niezawodność systemu bazodanowego, tak jak w przypadku Smart Grids.

Zastosowanie takich mechanizmów daje możliwość synchronicznej, konfigurowalnej poprzez API replikacji danych pomiędzy węzłami klastra w momencie, gdy dane są zatwierdzane na jednym z węzłów. Pozwala to osiągnąć bardzo wysokie bezpieczeństwo danych na wypadek awarii.

Stosując mechanizmy relacyjnych baz danych typu MySQL, które pomagają w systematyzacji danych, można dostrzec sporą stratę wydajności przy przetwarzaniu danych. Aby wyjść naprzeciw tym problemom powstały mechanizmy zakładające zapisywanie danych w formacie Klucz – Wartość (ang. *Key – Value*), dając początek nierelacyjnym bazom NoSQL czyli Not Only SQL (z ang. nie tylko SQL) które pełnią często rolę tzw. *cache* dla systemów informatycznych. Konsekwencją odrzucenia modelu relacyjnego jest to, że dane przechowywane w bazie danych nie wymagają ściśle określonych schematów. Często nie ma w nich złączeń relacyjnych, dzięki czemu możliwe jest łatwe skalowanie danych w poziomie, a co za tym idzie realizacja zapytań jest dużo bardziej efektywna. Model ten jest wykorzystywany w przypadku przetwarzania olbrzymich ilości danych, do których często się odwołujemy. W przypadku aplikacji tworzonych na potrzeby Smart Grids danymi, do których będziemy się musieli często odwoływać będą dane zawierające odczyty i parametry techniczne sieci. To od szybkości ich przetwarzania będą zależały szybkość decyzji, a co za tym idzie utrzymania sprawności systemu energetycznego. Istnieje wiele implementacji tego rozwiązania, które pozwoliły ich twórcom znacznie zwiększyć efektywność systemów takich jak Google czy Facebook. Choć więc poprawić efektywność systemu warto zastosować mechanizm NoSQL – Redis. Warto wybrać ten mechanizm przez wzgląd na wiele zalet którymi się charakteryzuje, a do których można zaliczyć maksymalną wartość danych pojedynczego wiersza sięgającą aż 512 MB danych oraz możliwość określenia czasu, przez jaki dane powinny być przechowywane w bazie danych Redis, nim zostaną z niej automatycznie usunięte. Redis zakłada przechowywanie i manipulację na danych w pamięci operacyjnej, co znacznie podnosi wydajność. Redis jest serwerem struktur danych, gdyż klucze mogą zawierać oprócz łańcuchów znaków tablice hashowe oraz listy i zbiory (w tym sortujące). Redis oferuje: atomowe operacje, transakcje, pub/sub, skryptowanie wewnątrz bazy za pomocą Lua i replikację typu Master-Slave. Mechanizmu Redis do szybkiego przetwarzania danych przy kooperacji z MySQL używa portal Flickr.

W budowanym modelu infrastruktury IT dla Smart Grids warto również wykorzystać inne efektywne otwarte rozwiązania informatyczne, jak chociażby system plików MooseFS (ang. *Moose File System*). Ten rozproszony sieciowy system plików sprawia, że systemy z niego korzystające stają się wysoko dostępne oraz odporne na awarie. Umożliwiają przechowywanie olbrzymich plików – maksymalnie o rozmiarze 128 PebiBajtów (2^{37} Bajtów) oraz tworzenie macierzy dyskowych o rozmiarze maksymalnym 16 EksbiBajtów (2^{64} Bajtów). Umożliwia on organizację danych w sposób podobny do środowisk Unixowych, zapewniając przy tym spore możliwości w przydzielaniu uprawnień, jak chociażby definiowanie ich dla adresów IP, czy zabezpieczanie danych hasłem dostępu. W tym systemie plików dane są rozpowszechniane na wiele maszyn wchodzących w skład instancji MooseFS. Pliki są dostępne dla każdej z maszyn połączonych sieciowo. Modyfikacje plików są przenoszone w ramach całego systemu. Przestrzeń dyskowa powstała na zasadzie połączonych maszyn jest widoczna dla użytkownika w formie jednego olbrzymiego zasobu. System MooseFS w połączeniu otwartym oprogramowaniem, jakim jest OpenNebula (platforma do przetwarzania w chmurze)

tworzą niezawodną chmurę obliczeniową (ang. *cloud computing*), która da spore możliwości obliczeniowe dla aplikacji Smart Grids.

W infrastrukturze IT na potrzeby Smart Grids nie może zabraknąć usług serwera WWW, które będą hostować (udostępniać) aplikacje użytkownikom systemu. Najbardziej znanym przedstawicielem grupy jest Apache, który posiada 65% udziału w rynku. Warto jednak zwrócić uwagę na inną nieco mniej popularną, lecz bardziej efektywną usługę – Nginx. W sytuacji olbrzymiej ilości wywołań aplikacji Smart Grids ze stacji klienckich, jakiej możemy się spodziewać, usługa Apache może znacznie stracić na efektywności działania, a nawet spowodować wyłączenie serwera. Opisany problem może wystąpić, gdyż każde z wywołań serwera Apache tworzące nową sesję tworzy również nowy wątek w systemie operacyjnym odpowiedzialny za daną sesję. Może się więc zdarzyć sytuacja awarii systemu jak w przypadku ataku DDoS (ang. *Distributed Denial of Service* – rozproszona odmowa usługi) czyli braku możliwości przetworzenia żądań klienckich przez serwer WWW z powodu zajętości wszystkich dostępnych zasobów. Nginx został zaprojektowany i stworzony pod kątem wysokiej dostępności na silnie obciążonych serwisach. Przy projektowaniu tego systemu wysoki nacisk położono również na skalowalność, niskie wymagania sprzętowe oraz minimalną ilość pochłanianych zasobów. Zaletą Nginx jest architektura, w której działa jeden proces główny i wiele procesów roboczych (pracujących w kontekście użytkownika nieuprzywilejowanego). Architektura ta jest odporna na wspomniane ataki DDoS. Kolejnymi zaletami są również: modułowość, zaawansowana obsługa zdarzeń, kolejkowanie, obsługa opóźnionego przyjmowania połączeń, obsługa szyfrowanych protokołów, współdziałanie z oprogramowaniem takim jak Keep-alive oraz możliwa asynchroniczność.

Trendy rozwoju aplikacji biznesowych w kierunku udostępniania ich za pomocą przeglądarek internetowych wynikające z chęci łatwiejszego sposobu ich propagacji, oraz unifikacji samych aplikacji na różnym sprzęcie spowodowały rozwój wielu otwartych technologii programistycznych do tworzenia dynamicznych aplikacji publikowanych poprzez WWW. Aby spełnić założenie skalowalności i uzyskać wysoką wydajność tworzonych aplikacji warto wykorzystać język PHP (ang. *Hypertext Preprocessor*) z dodatkiem PHP-FPM, znakomicie współpracującym z wybranym powyżej serwerem WWW – Nginx. PHP jest językiem skryptowym mającym wielorakie zastosowanie do tworzenia aplikacji webowych. Jest szczególnie popularnym językiem w środowisku Linux, ale dostępnym również na inne platformy systemów operacyjnych. Jest to stosunkowo prosty, a zarazem wydajny język programowania aplikacji, który swoją składnią przypomina języki C oraz Perl. Zaletą PHP jest możliwość współpracy z wieloma bazami danych. Najczęściej PHP jest wykorzystywany w połączeniu z bazą danych MySQL oraz serwerem Apache. Takie „trio” osadzone w środowisku Linux określa się mianem platformy LAMP. Dodatek PHP-FPM jest to *patch* zawarty w jądrze PHP, który pozwala zarządzać procesami interfejsu programistycznego FastCGI. Interfejs ten przyspiesza aplikacje internetowe oparte na technologii CGI (ang. *Common Gateway Interface*), oraz zapewnia ich lepszą skalowalność. Technologia CGI to interfejs, umożliwiający komunikację pomiędzy aplikacjami a oprogramowaniem serwera hostującego strony WWW umożliwiającą dynamiczne generowanie kodu HTML po stronie serwera (np. pobieranie danych z bazy danych i wyświetlanie ich na stronie WWW). Zastosowanie FastCGI oraz PHP-FPM zmniejsza zużycie pamięci, poprawia wydajność poprzez

lepsze zarządzanie procesami PHP, poprawia zarządzanie uprawnieniami PHP, podnosi dostępność gdyż awaria PHP nie powoduje błędów w działaniu serwera Nginx.

Wymaganie wysokiej interoperacyjności przy jednoczesnym zachowaniu wysokiego poziomu bezpieczeństwa danych, wymusza stosowanie wydajnych, bezpiecznych, a zarazem uniwersalnych i otwartych w sensie technologii mechanizmów wymiany danych. Mechanizmy te będą na ogół wykorzystywane do usystematyzowania i uporządkowania danych, jakimi będą się wymieniały systemy Smart Grids, a także będą zapewniały bezpieczny ich transport. Najbardziej rozpropagowaną technologią wymiany danych spełniającą opisane wymagania jest technologia WebServices oparta głównie o wymianę danych w formacie SOAP + XML poprzez protokół http/https. Technologia ta zakłada walidację przesyłanych komunikatów za pomocą zdefiniowanych WSDL. Interfejs ten ma poważną konkurencję w nieco mniej powszechnej jeszcze technologii RESTful, której idea jest komunikacja z serwerem poprzez wywoływanie odpowiednio sformatowanych adresów WWW przekazując odpowiednio sformatowane dane w standardzie JSON. Zastosowanie JSON, który ma pełne wsparcie w PHP, zmniejsza znacznie ilość przesyłanych danych w komunikacie w stosunku do struktury XML. Posługiwanie się oboma typami interfejsów z przez programistów PHP znacznie ułatwi wykorzystanie Zend Framework. Z punktu widzenia bezpieczeństwa danych istotnym jest, aby oba opisane typy interfejsów API były udostępniane za pomocą zabezpieczonego protokołu https (ang. Hypertext Transfer Protocol Secure). Dla zapewnienia maksymalnej uniwersalności aplikacje Smart Grids powinny obsługiwać oba standardy wymiany danych.

Ze względu na dane gromadzone i przetwarzane przez systemy Smart Grids, bardzo istotnym ich elementem jest zapewnienie wysokiego poziomu bezpieczeństwa. Oprócz stosowania wspomnianych już zabezpieczeń danych w systemie poprzez ich replikację, wykonywane kopii zapasowych, czy stosowanie szyfrowanych certyfikatami SSL połączeń systemy Smart Grids powinny być dodatkowo zabezpieczone. Do podstawowych zabezpieczeń jakie powinny być stosowane w systemach tego typu powinny należeć zapory *firewall* z określeniem reguł routingu dla ruchu wchodzącego i wychodzącego. Zapory te mają na celu eliminowanie niepożądanego ruchu w sieci. System operacyjny Linux Debian ma wbudowaną zaporę systemową *firewall*, którą można konfigurować za pomocą rozbudowanych reguł w iptables.

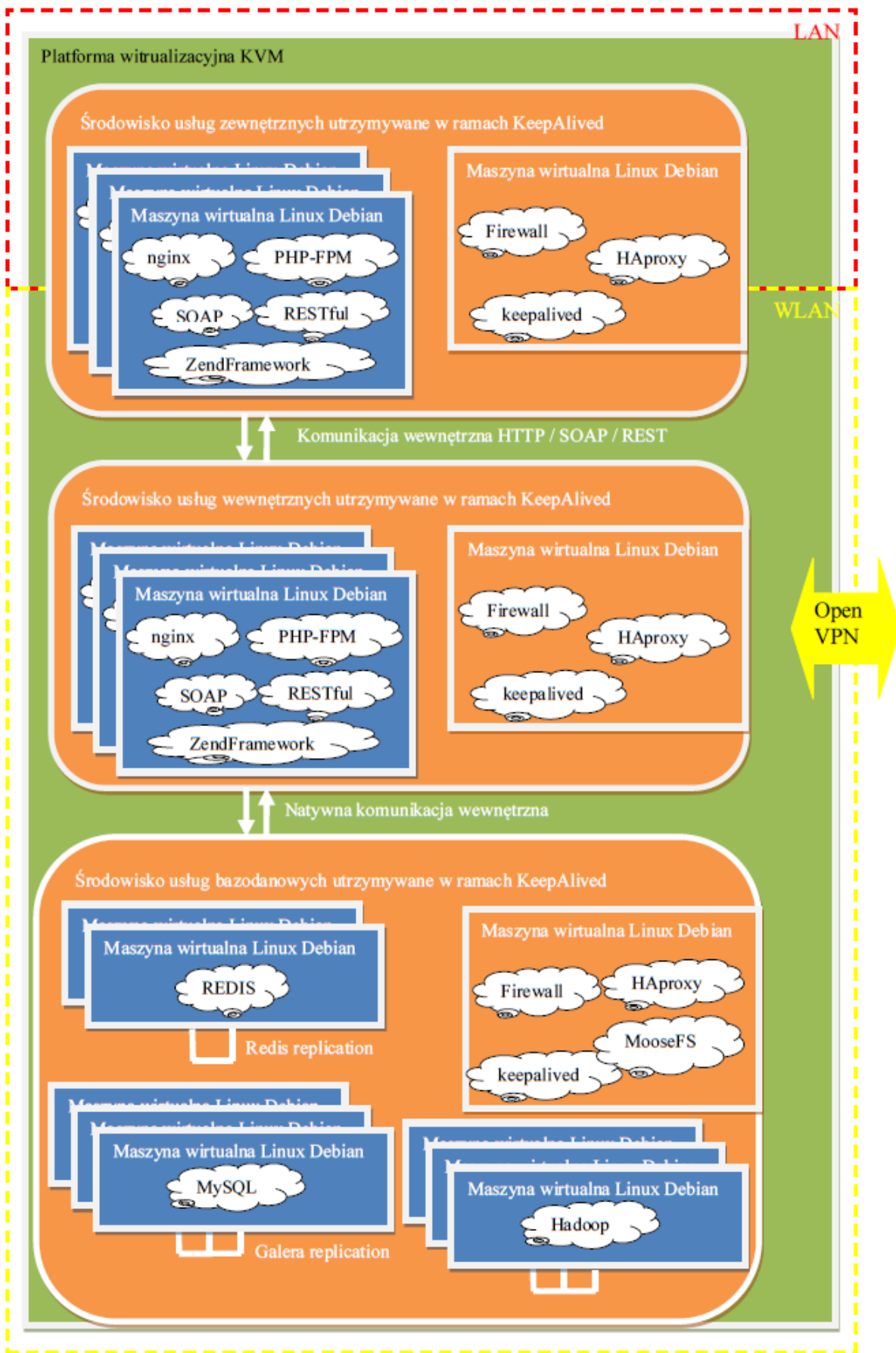
W przypadku systemów rozproszonych, do których należą systemy Smart Grids, niebagatelnym elementem zabezpieczeń jest stosowanie tunelowanego ruchu pomiędzy wyspami systemu informatycznego tworzącego tzw. wirtualną sieć prywatną – VPN (ang. *Virtual Private Network*). Tunelowanie ruchu pakietów danych poprzez stosowanie sieci VPN daje możliwość połączenia znajdujących się w różnych lokalizacjach klientów końcowych, poprzez sieć publiczną, w taki sposób, że poszczególne węzły sieci publicznej są przezroczyste dla pakietów, a klienci końcowi współpracują w ramach jednej wirtualnej sieci prywatnej. Dla zapewnienia większego bezpieczeństwa i podniesienia efektywności transferu dane przesyłane kanałami VPN mogą być szyfrowane oraz kompresowane. Do budowy sieci VPN warto użyć otwartego oprogramowania OpenVPN. Pozwala ono na

tworzenie szyfrowanych połączeń punkt – punkt lub strona – strona. Do tworzenia szyfrowanych połączeń OpenVPN wykorzystuje biblioteki OpenSSL oraz protokoły SSLv3/TLSv1. W ramach OpenVPN istnieje wiele metod uwierzytelnienia użytkowników takich jak: klucze dostępu, certyfikaty lub poprzez poświadczenia nazwy użytkownika i hasła.

Chcąc spełnić postawione założenia SLA należy zwrócić uwagę na usługi, które podnoszą poziom dostępności usług systemu IT dla Smart Grids i będą balansowały jego obciążenie, czy przekierują zapytania na inne serwery w razie awarii serwerów podstawowych.

Jedną z takich usług o bogatym spektrum zastosowań jest HAproxy. Jest to darmowe, ultra szybkie i niezawodne rozwiązanie oferujące wysoką dostępność (ang. *High Availability*), *load balancing* i *proxy* dla aplikacji opartych o TCP/IP. Szczególnie nadaje się do najbardziej obciążonych stron internetowych, kiedy wymagana jest pełna dostępność. HAproxy potrafi obsłużyć jednocześnie dziesiątki tysięcy połączeń, w pełni wykorzystując możliwości sprzętu. Dodatkowo w HAproxy można konfigurować ACL (ang. *Access Control List*), czyli zbudować usługę *firewall* dla aplikacji, filtrując ruch po IP, UserAgentach czy nawet ustawionych *cookies*. *Load Balancer* w HAproxy potrafi jednocześnie obsługiwać wiele domen. Usługa ta posiada wsparcie dla szyfrowania transmisji poprzez SSL oraz wiele innych ciekawych funkcji. Kolejną usługą współpracującą z HAproxy przy zapewnieniu wysokiej dostępności systemu IT dla Smart Grids jaką wykorzystamy w budowanym modelu systemu będzie usługa *Keepalived*. Usługa ta wykorzystywana jest do monitorowania fizycznych serwerów klastra Linux Virtual Server (LVS). Usługę można skonfigurować w taki sposób, aby z puli klastra usuwała nie działające fizyczne serwery (które przestały odpowiadać) oraz informowała administratora za pomocą poczty elektronicznej o zaistniałym problemie. Dodatkowo, w celu przeniesienia zasobów katalogowych w sytuacjach awaryjnych, *keepalived* używa niezależnego protokołu routingowego *Virtual Router Redundancy Protocol (VRRPv2)*.

Przetworzenie danych zgromadzonych przez system Smart Grids będzie wymagało skorzystania z specjalizowanych systemów. W tym celu można skorzystać z systemu Hadoop. Jest to platforma do przechowywania i zarządzania bardzo dużymi zbiorami danych, która daje się skalować. Koncepcja działania Hadoopa polega na podziale dużych zbiorów na mniejsze fragmenty, które mogą być równolegle przetwarzane w wielu węzłach wykorzystujących standardowe serwery. Hadoop nie ma zastępować istniejącej infrastruktury centrów danych, a uzupełniać ją o nowe funkcje związane przede wszystkim z możliwością analizy dużych zbiorów danych strukturalnych i niestukturalnych. Wykorzystując narzędzia, takie jak Apache Flume (obsługujący dwukierunkową wymianę danych między systemami RDBMS i Hadoop) czy Apache Sqoop (przekazujący logi systemowe do Hadoop w czasie rzeczywistym), można zintegrować istniejący system z oprogramowaniem Hadoop i przetwarzać dane bez względu na ich objętość. Wzrost liczby danych będzie jednak wymagał dodania kolejnych węzłów Hadoop obsługujących rosnącą pamięć masową. Stan ten będzie także wymagał zwiększenia liczebności serwerów przetwarzających dane. W instalacjach Hadoop warto skorzystać z dodatku Apache Hive, który jest interfejsem SQL dla Hadoopa.



Rys. 1. Projekt modelu infrastruktury IT możliwy do wykorzystania w systemie sieci inteligentnych Smart Grids

Zestawiając odpowiednio wszystkie wspomniane komponenty otwartego oprogramowania można zbudować kompletny model infrastruktury IT na potrzeby Smart Grid. Przykład takiego modelu infrastruktury IT przeznaczonego dla systemu Smart Grids przedstawiono na rys. 1. Prezentowany model zakłada, że główną platformą wirtualizacji będzie KVM. Na tej platformie zrytualizowane zostaną redundantne maszyny z systemem operacyjnym Linux Debian. W ramach platformy witalizacyjnej zostaną wydzielone trzy główne środowiska w ramach podsięci:

- a) środowisko systemów relacyjnych baz danych i baz NoSQL dostępne jedynie z sieci wewnętrznej poprzez usługi,
- b) środowisko usług WebService i aplikacji wewnętrznych udostępniające usługi na potrzeby wewnętrzne,
- c) środowisko usług WebService i aplikacji zewnętrznych pośredniczące w komunikacji pomiędzy sieciami LAN a WLAN udostępniające usługi na zewnątrz.

Zarówno środowisko usług WebService i aplikacji wewnętrznych jak i środowisko usług WebService i aplikacji zewnętrznych będą utrzymywane w ramach usług KeepAlived, które będą gwarantowały ich dostępność, a w razie potrzeby będą raportować o zaistniałych problemach i przekierowywać ruch do dostępnych maszyn redundantnych. Środowiska te będą miały odpowiednio skonfigurowane reguły zabezpieczeń *firewall* w iptables oraz reguły *load balancingu*, *proxy*, *ACL* i *SSL* skonfigurowane w HAproxy. Aplikacje webowe stworzone w ramach PHP-FPM z dodatkiem ZendFramework, oraz usługi WebService wykorzystujące technologie SOAP oraz RESTful zostaną osadzone na serwerach WWW Nginx. W środowisku bazodanowym zostanie skonfigurowany system plików MooseFS. Dostępność tego środowiska będzie gwarantowana usługami KeepAlived. Zaś ochroną tego środowiska zajmie się *firewall* z odpowiednimi regułami. Równoważenie obciążenia, oraz zarządzanie regułami ruchu aplikacji będzie spoczywało na usłudze HAproxy. Główną bazą relacyjną będzie klastrowy system MySQL – wsrep, który zagwarantuje bezpieczeństwo danych, oraz umożliwi podział funkcjonalny elementów klastra na *node*y zapisujące i odczytujące. Buforowaniem danych i ich szybką obróbką zajmie się system bazodanowy typu NoSQL – Redis. Jako mechanizm hurtowni danych zostanie wykorzystana baza danych NoSQL – Hadoop z usługami: Apache Flume, Apache Sqoop, Apache Hive. Bezpieczna komunikacja pomiędzy kolejnymi rozproszonymi lokalizacjami modelowanego systemu Smart Grids zostanie zabezpieczona usługą Open VPN.

Wnioski

Przeprowadzone w artykule analizy właściwości systemów IT opartych o otwarte oprogramowanie wskazuje na możliwość zbudowania wydajnej, skalowanej, wielodostępowej, bezpiecznej infrastruktury IT na potrzeby Smart Grids. Taką infrastrukturę można zbudować w

oparciu o otwarte oprogramowania stosowane obecnie przez światowych liderów rynku IT. Oznacza to, że znacząco można ograniczyć nakłady inwestycyjne na budowę i wdrożenie tego typu systemu w elektroenergetyce.

Konstruując taki system informatyczny dla elektroenergetyki warto wykorzystać doświadczenie liderów branży IT, którzy swoje rozległe systemy zbudowali w oparciu o otwarte technologie informatyczne. Użytkownicy otwartych systemów informatycznych na skalę masową przekonali się, że gwarantują one stabilną pracę systemów, pozwalają skalować ich rozmiar, a co najważniejsze zapewniają wysoki poziom bezpieczeństwa. Czynnikiem ten będzie jednym z kluczowych w procesie wdrażania infrastruktury inteligentnego opomiarowania.

Toczące się obecnie w Polsce dyskusje dotyczące określenia niezbędnych wymagań w zakresie interoperacyjności współpracujących ze sobą elementów sieci Smart Grid oraz elementów sieci domowych współpracujących z siecią Smart Grid” [5], powinny bardziej koncentrować się na kwestii zastosowania otwartych technologii informatycznych w tworzeniu Smart Grids. Ta technologia informatyczna może zasadniczo przyczynić się do wzrostu dynamiki rozwoju sieci inteligentnych i sfery oferowanych usług.

LITERATURA

- [1] Dyrektywa Parlamentu Europejskiego i Rady 2009/72/WE dotycząca wspólnych zasad rynku wewnętrznego energii elektrycznej
- [2] Dyrektywa Parlamentu Europejskiego i Rady 2009/73/WE dotycząca wspólnych zasad rynku wewnętrznego gazu ziemnego
- [3] Stanowisko Prezesa URE w sprawie niezbędnych wymagań wobec wdrażanych przez OSD E inteligentnych systemów pomiarowo-rozliczeniowych z uwzględnieniem funkcji celu oraz proponowanych mechanizmów wsparcia przy postulowanym modelu rynku. Warszawa, maj 2011
- [4] Silberchatz A., Galvin P. B., Gagne G., *Podstawy Systemów operacyjnych*, PWN 2005
- [5] Stanowisko Prezesa URE w sprawie niezbędnych wymagań dotyczących ram interoperacyjności współpracujących ze sobą elementów sieci Smart Grid oraz elementów sieci domowych współpracujących z siecią Smart Grid. Warszawa, styczeń 2013
- [6] HANs within Smart Grids. GEF Energy Efficiency Project TF No. 054104. A.T. Kearney, Warszawa 2012.
- [7] Skomudek W.: Smart Grids building in Poland. PolSCA Meeting „Smart Energy Networks”. Brussels, 5th may 2011.
- [8] Power Vision 2050. Scenarios „Conservative” and „Progressive”. EUREL Task Force 2011.

Michał PUDEŁKO¹, Waldemar SKOMUDEK²
CROSS sp. z o.o., 45-061 Opole, ul. Katowicka 39 (1)
Politechnika Opolska, Katedra Zarządzania Projektami, 45-067
Opole, ul. Ozimska 75 (2)