

OpenLaszlo

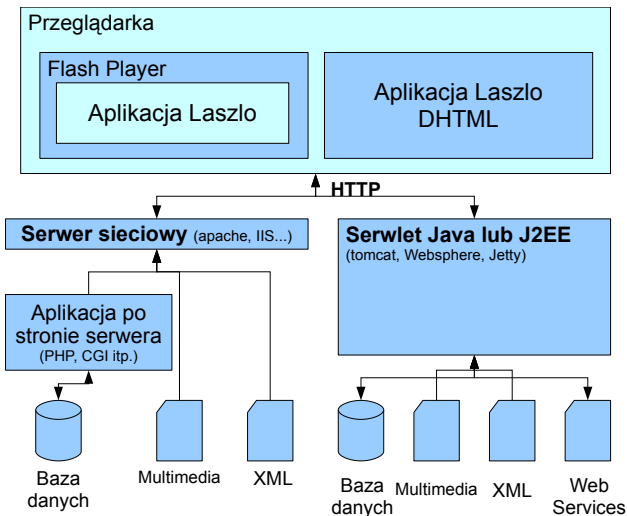
Spis Treści

- 1 **OpenLaszlo**
 - Co to jest?
 - Historia
 - Idea
 - Architektura
 - Jako Flash lub DHTML
 - Jako servlet lub SOLO
 - Jak to działa?
 - Język LZX
 - Struktura programu
 - Skrypty
 - Obiekty i klasy
 - Atrybuty i metody
 - Więzy
 - Kto tego używa
 - Konkurencja
 - Podsumowanie

Otwarta platforma oparta na licencji CPL przeznaczona do budowy bogatych aplikacji internetowych (RIA). Posiada bogate wsparcie dla grafiki, w tym wektorowej, animacji, budowy interfejsu użytkownika itd.

- 2001** Początek pracy nad Laszlo Presentation Server (LPS).
- 2002** Pierwsza wersja (używana w projekcie Behr)
- 2004** Udostępniony jako OpenSource
- 2005** Nazwa zmieniona na OpenLaszlo
- 2007** OpenLaszlo 4 (Flash i DHTML)

- Umożliwienie łatwego tworzenia aplikacji które działają i wyglądają tak samo niezależnie od systemu operacyjnego (Windows, Linux), przeglądarki (IE, Firefox, Safari) czy technologii (Flash, DHTML...).
- Aplikacje mają być proste i przyjazne dla użytkownika
- Brak konieczności instalacji dodatkowego oprogramowania
- Otwarta platforma, bez haczyków



Jako Flash lub DHTML

Klient

OpenLaszlo został zaprojektowany aby wspierać tworzenie aplikacji Ajax'owych na wiele platform oraz urządzeń. Wersja 3 wspiera Flash 7 w górę, wersja 4 dodaje wsparcie dla DHTML.

Komunikacja aplikacji z serwerem zawsze odbywa się za pomocą Ajaxa, więc aplikacja po stronie serwera nie musi być modyfikowana aby współpracować z nowym klientem.

Flash a DHTML

Flash nie obsługuje nieskompilowanego JavaScriptu, dlatego kompilator OpenLaszlo tłumaczy go do bytekodu. Z drugiej strony wersja Flash pozwala na wstawki multimedialne.

DHTML z kolei pozwala na pisanie kodu w Javascript, więc funkcjonalność obu wersji jest (prawie) identyczna.

Jako servlet lub SOLO

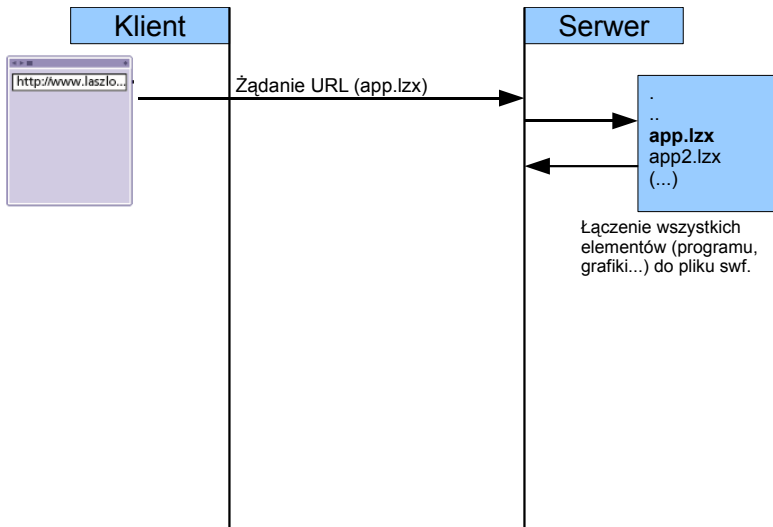
Zadanie OpenLaszlo

OpenLaszlo działa także jako proxy pomiędzy naszą aplikacją napisaną w LZX a aplikacją po stronie serwera. Gdy jest to konieczne kompiluje kod aplikacji i wysyła do przeglądarki odpowiednią wersję (obecnie Flash lub DHML).

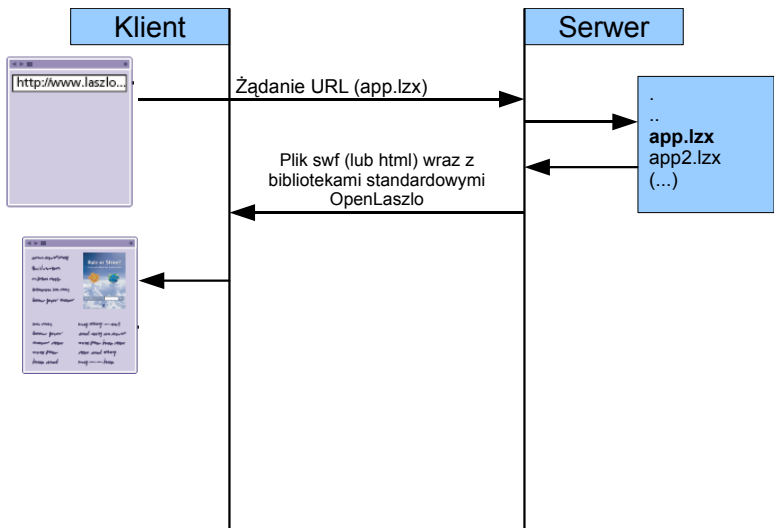
Serwer

Jeżeli nie chcemy korzystać z serwera Javowego możemy skompilować kod LZX do wersji SOLO (Standalone OpenLaszlo Output). Taka aplikacja może potem być używana z dowolnym innym serwerem (Apache, Lighttpd, IIS...) oraz językiem po stronie serwera (PHP, Python, CGI).

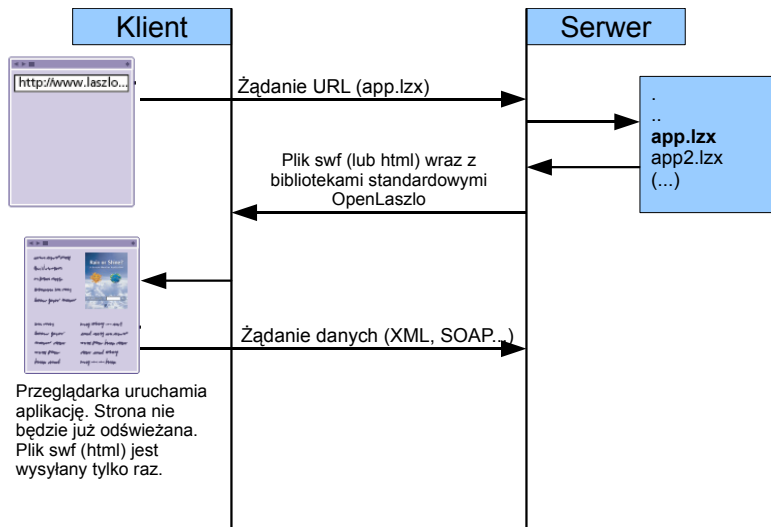
Jak to działa?



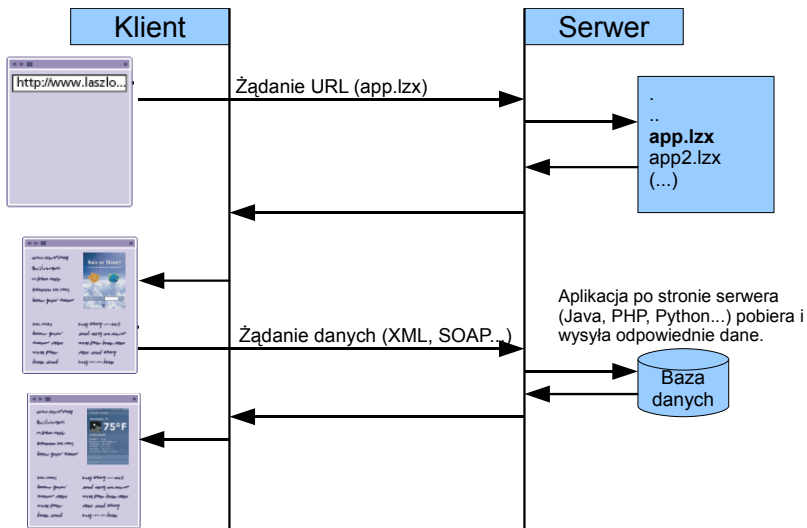
Jak to działa?



Jak to działa?



Jak to działa?



Przykładowa aplikacja

Przekładowa aplikacja

```
<canvas width="500" height="350">
<window width="200" height="250" title="Window" resizable="true">
  <simplelayout axis="y" spacing="10"/>
  <text>Here is some text.</text>
  <text>I could ramble for hours.</text>
</window>
</canvas>
```

Struktura programu

Deklaratywny język

Aplikacje OpenLaszlo tworzone są w sposób deklaratywny. Całość programu jest zawarta w elemencie canvas który odpowiada widocznej części aplikacji. Wewnątrz niego można umieszczać pozostałe elementy (tekst, przyciski, widoki), a także je zagnieżdżać.

Każdy element ma pewien zestaw atrybutów które pozwalają na zmianę jego wyglądu i działania.

Skrypty

Skrypty w aplikacji można umieścić w tagach `<script>` (mogą wystąpić tylko bezpośrednio wewnątrz `<canvas>`). Kod taki jest wykonywany od razu po uruchomieniu aplikacji.

Przykładowy skrypt

```
<canvas width="500" height="200" debug="true">  
  <script>  
    Debug.write("Hello, World!");  
  </script>  
</canvas>
```

Obiekty i klasy

Deklaracja klasy i obiektu

```
<canvas width="500" height="80">
  <class name="MyClass" width="80" height="25" bgcolor="#CFD9AB">
    <text align="center" valign="middle">Hello, World!</text>
  </class>

  <MyClass name="myFirstInstance"/>
</canvas>
```


Atrybuty

Atrybuty

```
<canvas width="500" height="80">
  <class name="SpecialButton" extends="button" onclick="changeLabel(">
    <attribute name="changeToLabel" value="Clicked!" type="string"/>
    <method name="changeLabel">
      this.setAttribute('text', this.getAttribute('changeToLabel'));
    </method>
  </class>

  <simplelayout axis="y" spacing="10"/>
  <SpecialButton changeToLabel="Thanks!">click me!</SpecialButton>
</canvas>
```

Więzy

Ciekawym rozwiązaniem wprowadzonym w OpenLaszlo są więzy (ang. Constraints). Pozwalają one uzależnić wartość jednego atrybutu od innych atrybutów.

Atrybuty

```
<canvas height="120">  
  <checkbox id="cb" text="Show Blue Square"/>  
  <view visible="{cb.value}" width="30" height="30"/>  
  <simplelayout/>  
</canvas>
```

Kto tego używa

- FNAC - Francuski sklep z elektroniką.
- Behr - pozwala na podgląd wyglądu pomieszczeń po pomalowaniu.
- Wal-Mart - jeden z największych sklepów internetowych.
- Glify - tworzenie diagramów online
- Worrydream - Strona domowa Breta Victora.

Konkurencyjne rozwiązania

- Adobe Flex

Konkurencyjne rozwiązania

- Adobe Flex
- Silverlight

Konkurencyjne rozwiązania

- Adobe Flex
- Silverlight
- Google Web Toolkit (GWT)
- Echo3

Konkurencyjne rozwiązania

- Adobe Flex
- Silverlight
- Google Web Toolkit (GWT)
- Echo3
- ZK Framework
- Morfik

Dlaczego Flash i HTML a nie Java

- Aplety Javy ciągle niewygodne w użyciu.
- Małe zapotrzebowanie rynku na aplety Javy.
- Java sprawdza się dobrze po stronie serwera, nie klienta.
- Środowisko uruchomieniowe OpenLaszlo może zostać rozszerzone (na razie jest to Flash i HTML, ale nic nie stoi na przeszkodzie dodania apletów)