

Interaktiv visualisering av IP-nätverk

Steve Eriksson
Linköpings universitet

Bakgrund

Telenor NOC

- Network Operations Center
- Karlskoga
- Övervakning och drift
 - IP-nät
 - Mobiltelefoninät
- Felavhjälpning

Existerande system

- Nätverkskartor i olika filformat
- Mängd verktyg från olika leverantörer
 - Svåra att integrera
- Egenutvecklade verktygsprogram
 - UNIX-baserade skalprogram
- Experiment med SVG-baserade nätkartor

Problemet

- IP-nätverket är mycket omfattande
 - Många noder och ännu fler förbindelser
 - Komplexa beroenden
 - Svårt att skapa mental modell


```
3 MET-DST +02:00 %CDP-4-NVLANMISMATCH:Na
4 MET-DST +02:00 %MGMT-5-LOGIN_FAIL:User
5/2N: Interface GigabitEthernet0/2 changed state to
6/2UPDOWN: Line protocol on Interface GigabitEthern
7/2MET-DST +02:00 %DTP-5-NONTRUNKPORTON
8/2MET-DST +02:00 %PAGP-5-PORTFROMSTP:Po
9/2PORTATTEMPT: Attempted to connect to RSHELL
10/2FLAP: GigabitEthernet0/1 relearning 7 addrs per min
11/2FLAP: GigabitEthernet0/1 relearning 6 addrs per min
12/2FLAP: FastEthernet0/35 relearning 6 addrs per min
13/2
```

Varför nätkartor?

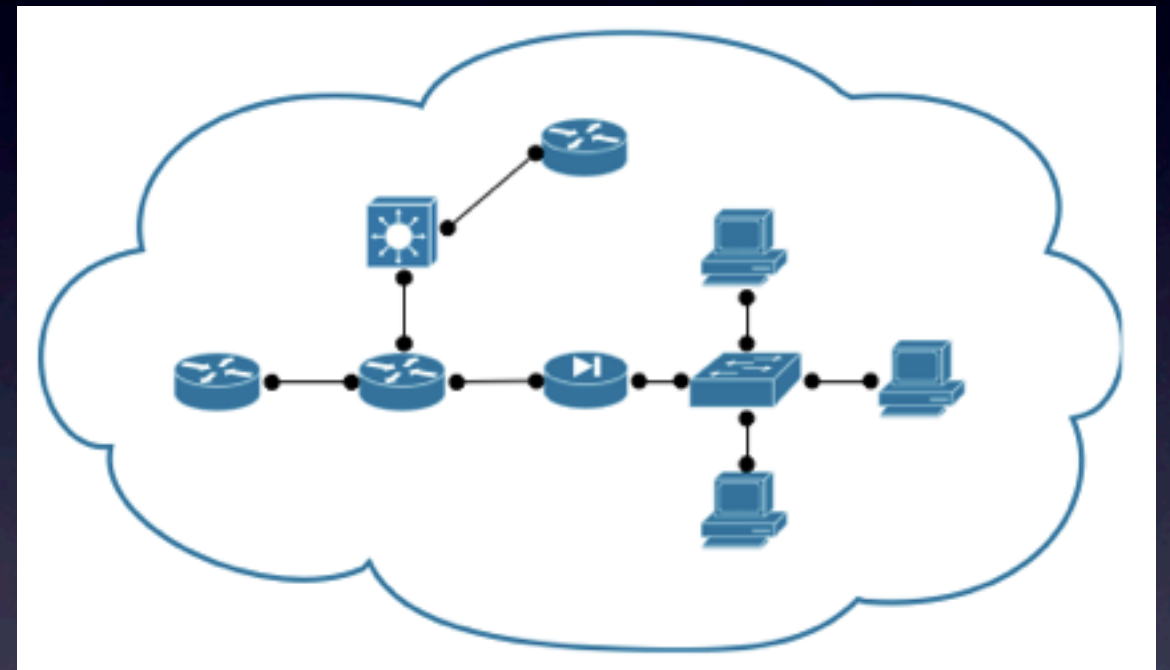
- Skapa övergripande bild av nätverket
- SVG
 - XML-baserat
 - Format för webben
 - Bindningar till JavaScript
 - HTML-dokument
 - Nätkarta och användargränssnitt

Syftet med arbetet

- Utveckla prototypsystem
- Nätverkskartor
 - SVG
 - Anropa befintliga verktygsprogram

Begreppsförklaring

- Nätverksvisualisering
 - Graf
- AJAX
- CGI
- XSLT
- GraphViz



Analys

Klient

1. Interaktivt SVG-dokument

- Dynamiskt förändra innehåll i SVG
- Bindning av JavaScript-funktioner

2. Fånga händelser i dokumentet

Server

3. Kommunikation mellan klient och server

- Anrop från klient
- "Exekvera nslookup för denna nod"

4. Behandling av anrop från klient

- Ta emot begäran
- Utför begäran
- Returnera resultatet

Designval

1. Bindning av JavaScript-funktioner

- Transformera dokument med XSLT

2. Hantering av användarinitierade händelser

- Expedieringsfunktion (dispatch)

3. Anrop från klient till server

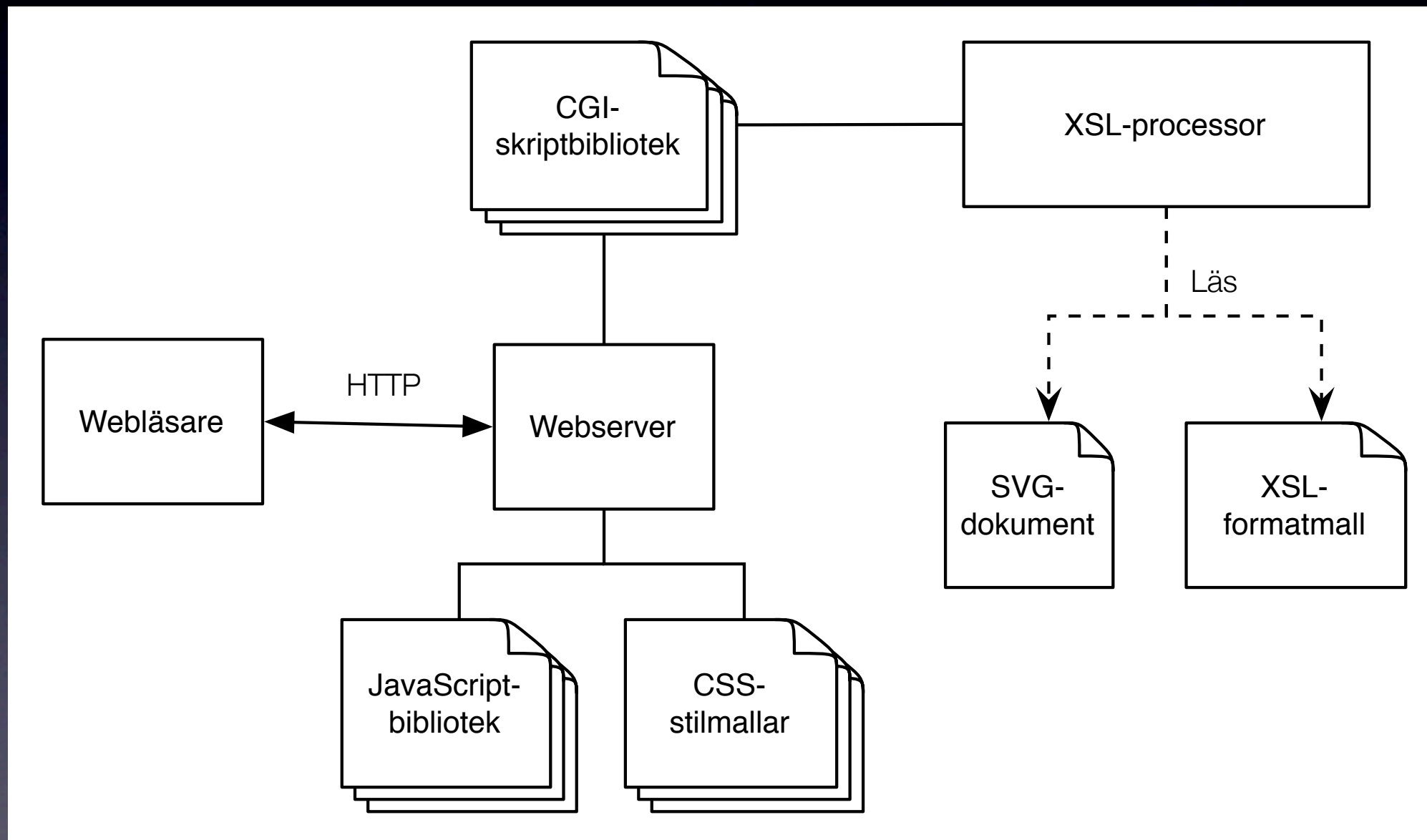
- XMLHttpRequest

4. Behandling av anrop från klient

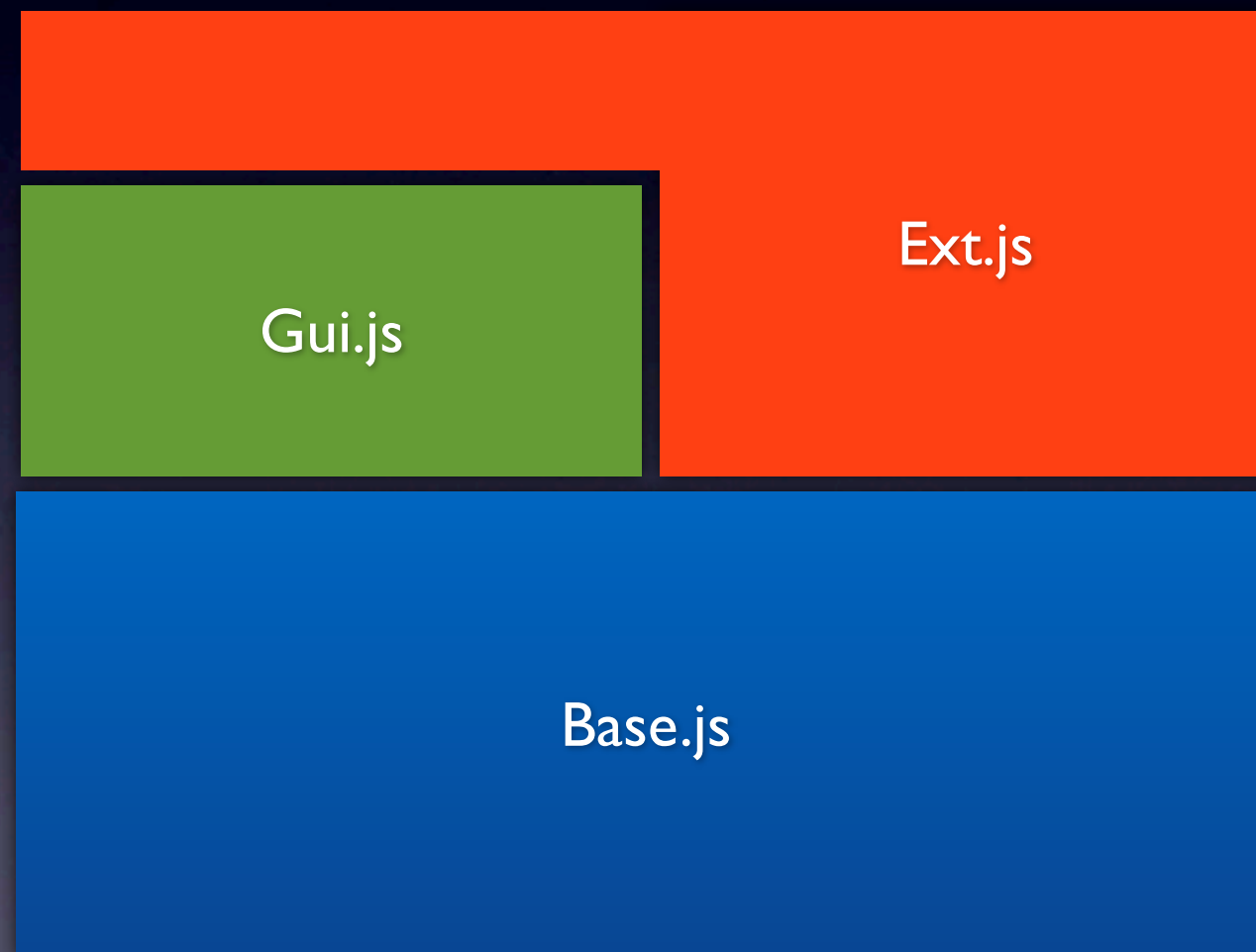
- Ett CGI-skript per anropad funktion

Implementation

Exekveringsvy



JavaScript-bibliotek



JavaScript-bibliotek

- Base.js
 - Funktioner bundna till händelser i SVG
 - Expedieringsfunktioner (dispatch)
 - Hjälpfunktioner
- Gui.js
 - Funktioner kopplade till det grafiska gränssnittet
- Ext.js
 - Kopplingar till externa program
 - Utökad funktionalitet

Expedieringsfunktion

```
base.makeDispatch = function() {  
  
    var funMap = {}; // function map  
  
    // Return the interface to the dispatch object  
    return {  
        addFunction: function( name, fun ) {  
            funMap[name] = fun;  
        },  
  
        removeFunction: function( name ) {  
            delete funMap[name];  
        },  
  
        listFunctions: function() {  
            var names = [];  
            for (name in funMap) {  
                if (funMap.hasOwnProperty( name )) {  
                    names.push( name );  
                }  
            }  
            return names;  
        },  
  
        handleEvent: function( evt ) {  
            for (name in funMap) {  
                if (funMap.hasOwnProperty( name )) {  
                    funMap[name]( evt );  
                }  
            }  
        }  
    }; // end return  
}
```

CGI-skriptbibliotek

- Program anropade av användare
 - Hämta IP-adress för nod
 - Ping
- Beräkning av SVG-dokument
 - Bindning till JavaScript-funktioner
 - Referenser till filer
 - Använder XSLT
 - CPAN - XML::LibXSLT

Skapa bindningar XSLT + Perl

```
use XML::LibXSLT;
use XML::LibXML;
use CGI;

my $parser = XML::LibXML->new();
my $xslt    = XML::LibXSLT->new();

my $source    = $parser->parse_file( $svg_file );
my $style_doc = $parser->parse_file( $xsl_file );

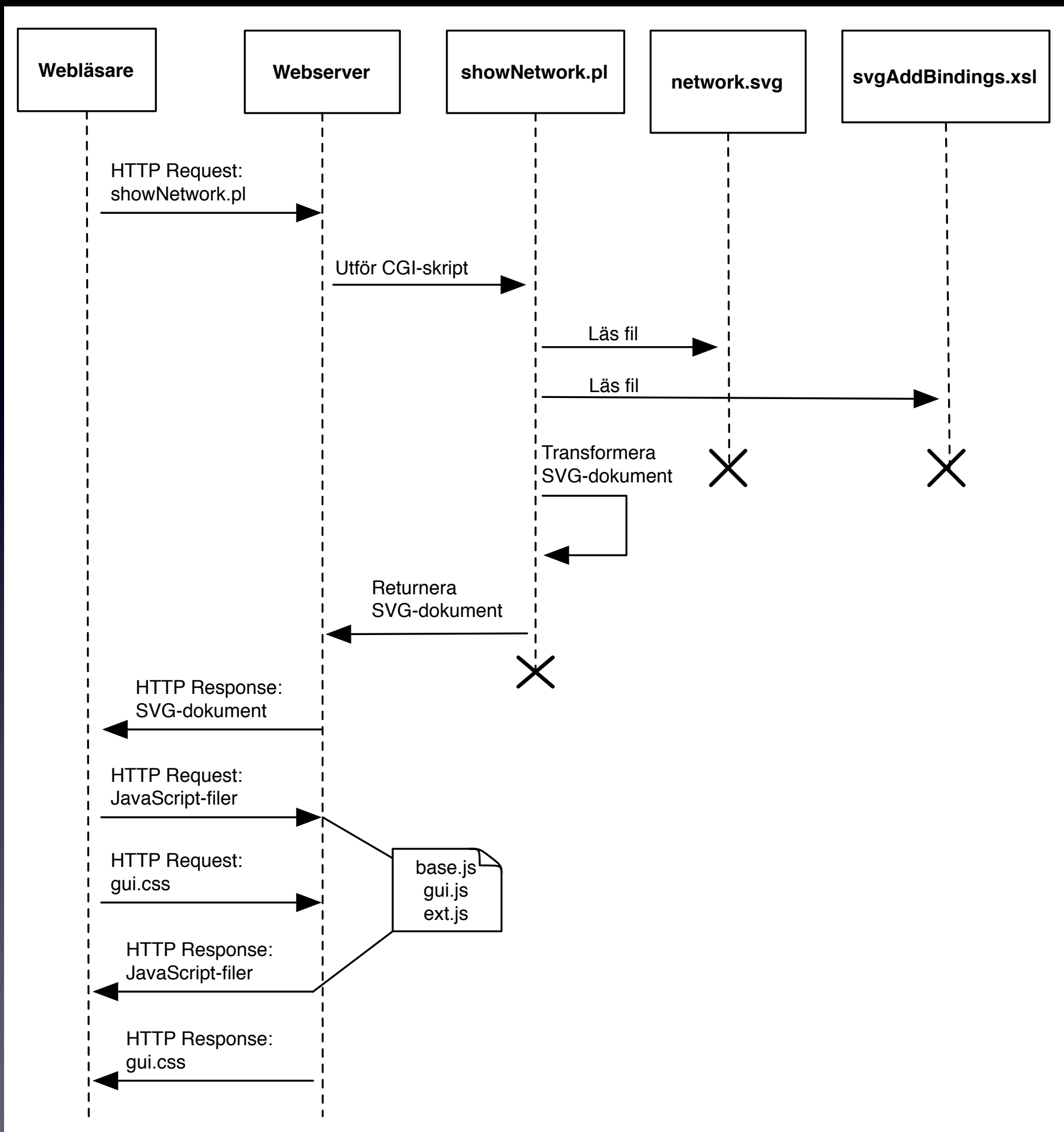
my $stylesheet = $xslt->parse_stylesheet( $style_doc );

my $result = $stylesheet->transform( $source );

print $cgi->header( "image/svg+xml" );
print $stylesheet->output_string( $result );
```


Generering av nätverkskarta

<http://vårdnamn.domän/cgi-bin/showNetwork.pl?network=namn.svg>



Exempel på berikat dokument

```
<g id="node3" class="node">  
<title>www.freebsd.org</title>  
<ellipse style="fill:lightgreen;stroke:lightgreen;" cx="1033" cy="-1144" rx="60.1041" ry="28.2843"/>  
</g>
```

```
<g onclick="node_onclick(evt)"  
  onmouseover="node_onmouseover(evt)"  
  onmouseout="node_onmouseout(evt)" id="node3" class="node" >  
<title>www.freebsd.org</title>  
<ellipse style="fill:lightgreen;stroke:lightgreen;" cx="1033" cy="-1144" rx="60.1041" ry="28.2843"/>  
</g>
```


Demonstration

Slutsats och diskussion

Resultat

- Prototypsystem
- Visualisera IP-nätverk
- Interaktiva nätverkskartor
- SVG som användargränssnitt

Öppen mjukvara och öppna standarder

- Komponenterna i system är gratis att använda
- All källkod är tillgänglig och kan undersökas
- Koden är ofta väldigt robust och väl testad
- Möjligt att byta ut komponenter
- Minskar risk för inlåsnings till en viss leverantör

Användbarhet

- Presentera resultat i SVG-format
 - Vektorgrafik
 - Klippa och klistra
 - Besvärligt att presentera text
 - Radbrytningar
 - Avstånd mellan tecken

Framtida arbete

- Grund för att bygga system för nätverksvisualisering
- Kompatibilitet med fler webbläsare
- Vidareutveckla funktionsmenyn
- Säkerhet
- Sammankoppling med verktygsprogram

Källkod

- <https://github.com/steeri/NetViz>

Frågor och kommentarer

- <https://github.com/steeri/NetViz>