

# Architektura

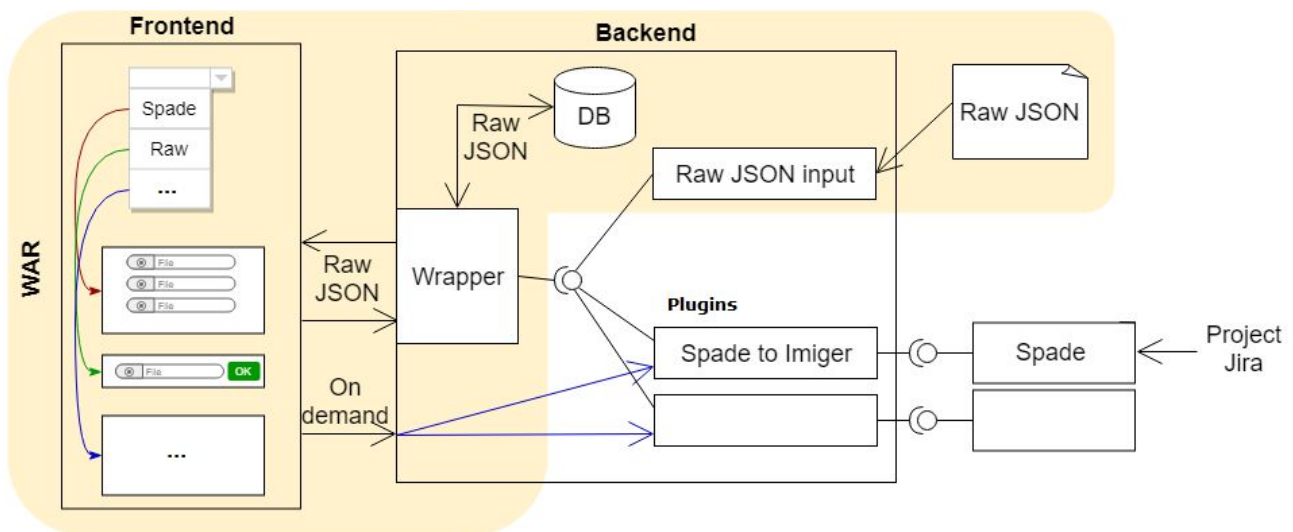
Dokument popisuje architekturu aplikace, především z pohledu přidané funkčnosti.

Autor: Patrik Harag

Verze: 1.0

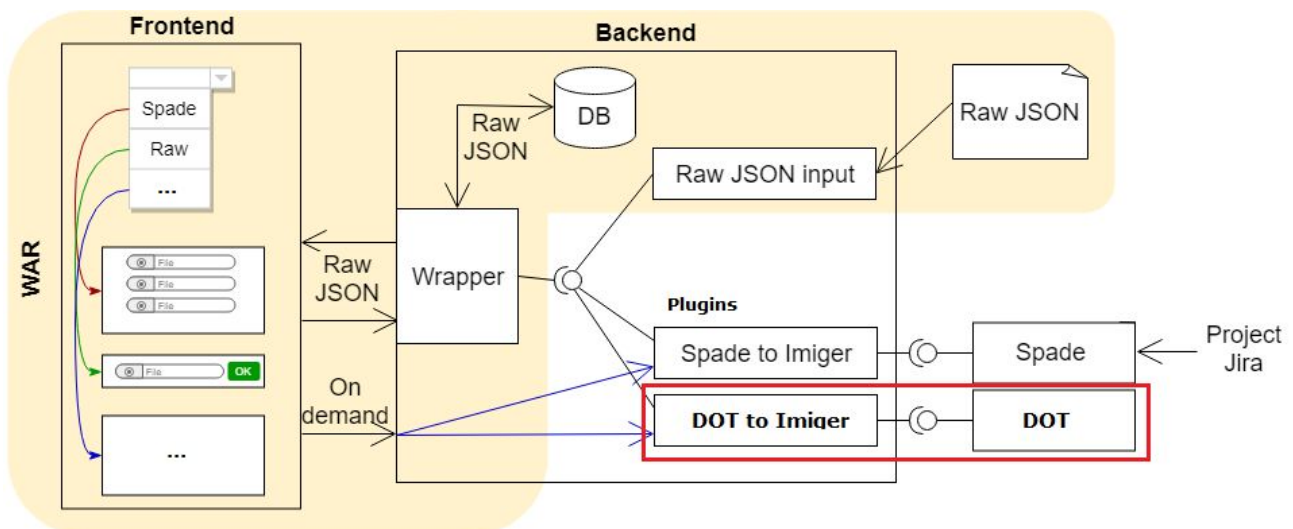
Datum: 2019-06-10

## Původní architektura



## DOT plugin

Hlavní změnou bylo přidání pluginu pro podporu DOT formátu. Došlo však i k drobným úpravám na backendu i frontendu, a to z důvodu zavedení podpory pro nový typ souboru (.dot) – do té doby byl vše pouze JSON a tomu odpovídaly i chybové hlášky, filtrování souborů, apod.

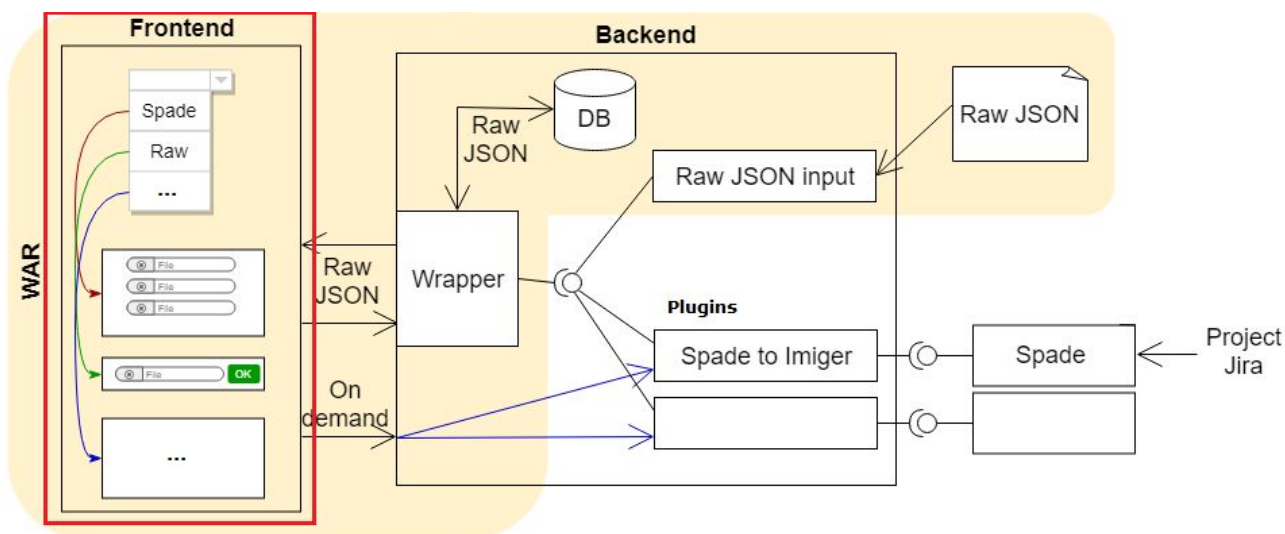


## Initial elimination

Cílem initial elimination bylo především vylepši přehlednost u velkých grafů. Vedlejším efektem je vylepšení responsivnosti aplikace. IE však nemá pozitivní vliv na prvotní načtení grafu - to je dáno způsobem, jakým se nakládá se skupinami. Vylepšení by znamenalo větší zásah do aplikace, se kterým nebylo počítáno a performance u IE ani nebyl klíčový požadavek.

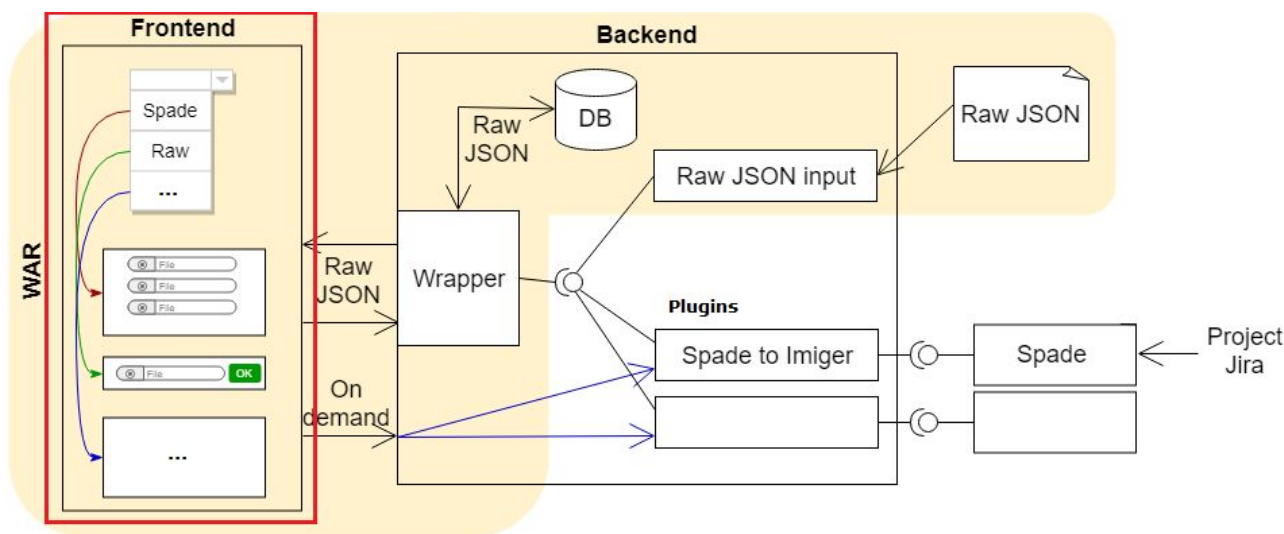
Initial elimination funguje tak, že pokud je počet viditelných prvků větší než 20, tak seskupuje uzly podle jejich archetypu. Pro každý archetyp je vytvořena skupina. Několik uzlů s nejvíce hranami je ponecháno mimo skupinu – proporcionálně podle jejich počtu, a to do maximálního počtu 20 zobrazených prvků. Se stávajícími skupinami nic neprovádí. Algoritmus je implementován v souboru *js/services/initialElimination.js*.

Došlo pouze ke změnám na frontendu.



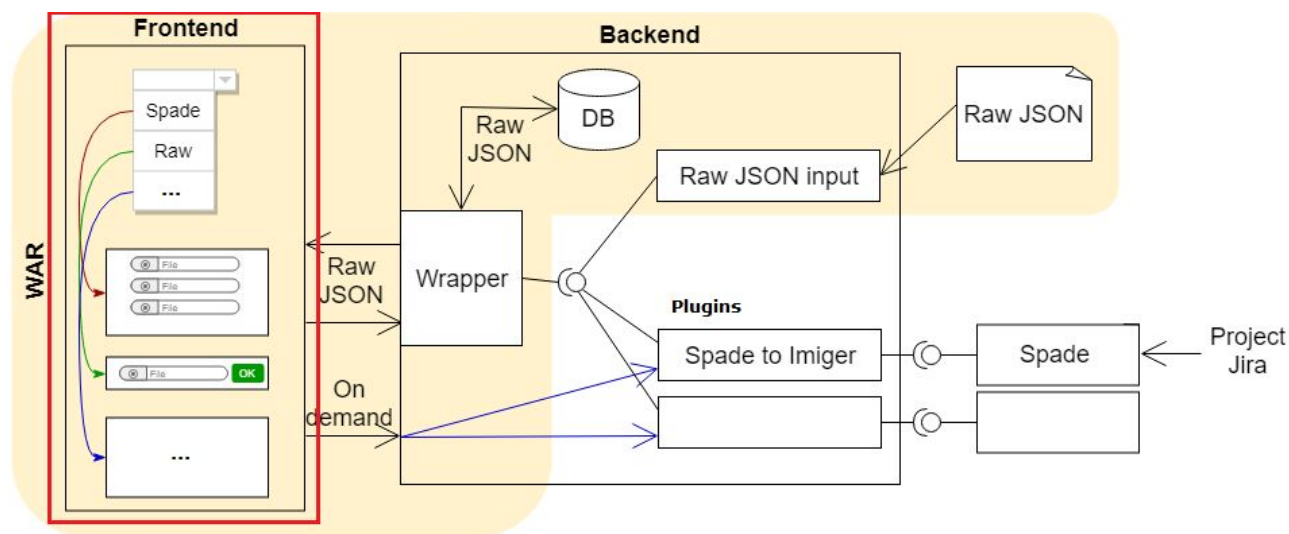
## Timeline

Do složky s JS soubory byla přidána složka *timeline* pro zobrazení časové osy. Timeline je do aplikace integrována jako další komponenta, nicméně se jedná o samostatnou aplikaci s vlastním datovým modelem.



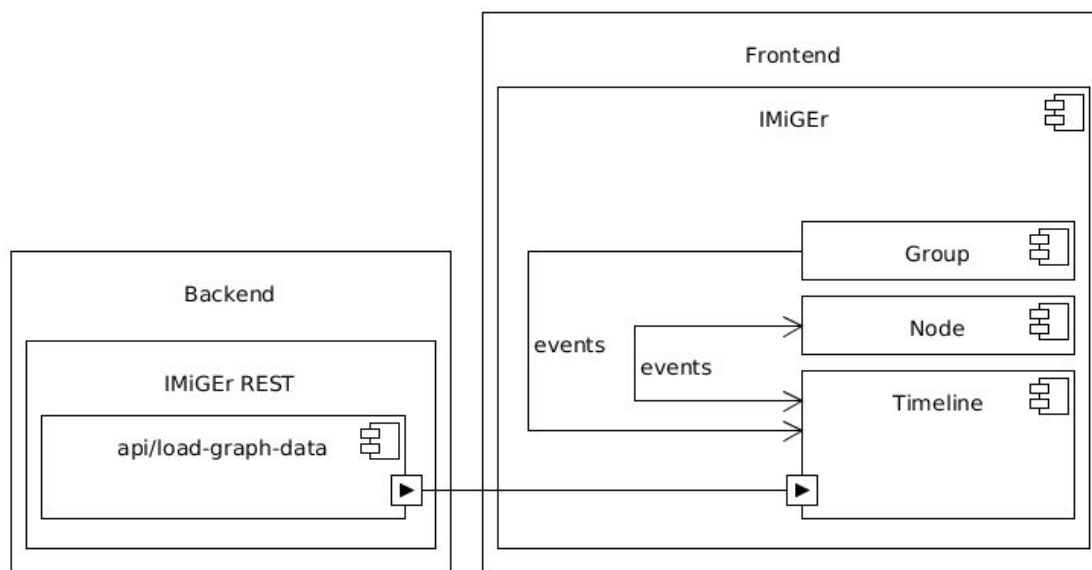
Součástí Timeline je i Bootstrap - framework obsahující návrhářské šablony. K načítání potřebných skriptů využívá Timeline requireJS kde hlavním zdrojovým souborem je main.js. V tomto souboru je načítání časové osy stejně jako zajištění načtení potřebných JS knihoven (jQuery, Bootstrap, noUiSlider, JQueryUI).

Obě aplikace (Timeline a Imiger) se mezi sebou dorozumívají pomocí událostí (např. *timelineClick*, *imigerExclude*, atd...). Implementace těchto událostí se nacházejí v souborech *group.js*, *node.js* pro IMiGER a *Timeline.js* pro časovou osu. Timeline aplikace navíc pro vizualizaci využívá data, která sama získává z REST API poskytovaného IMiGERem. O získání a následné předzpracování se stará skript *RestSource.js*.

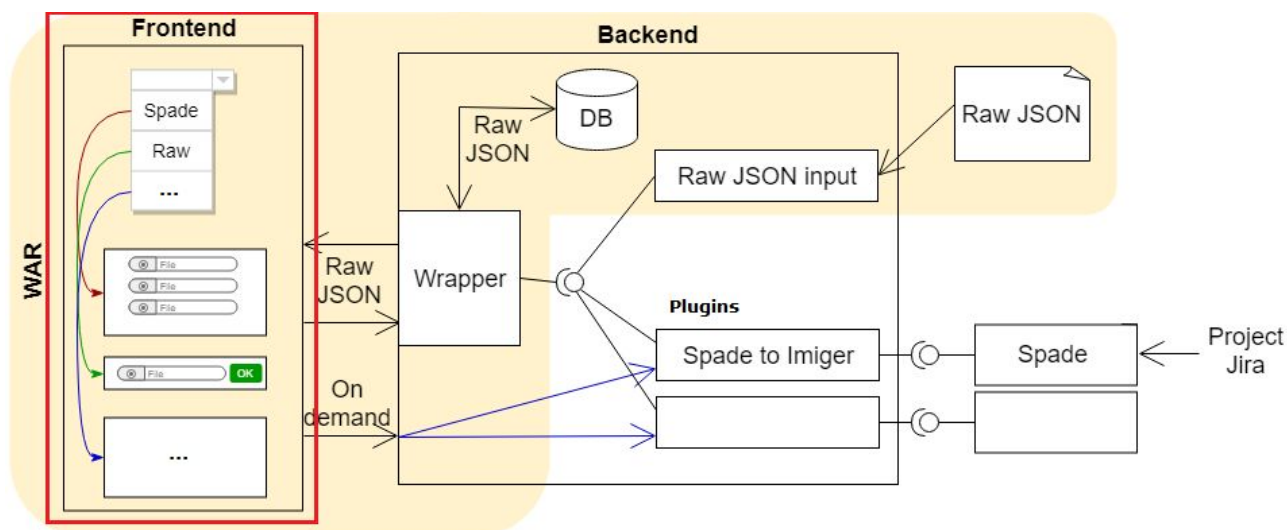


## Time range slider

Slider je použit při výběru vrcholů pro specifikaci rozmezí datumů. Do složky s JS soubory byla přidána složka se soubory pro zobrazení slideru (*js/timeline/auxiliary/*). Na slider byla vybrána JS knihovna noUiSlider, která je vysoce přizpůsobitelná. Zobrazení slideru je realizované v souboru *filterModalWindow.js*.



## Výběr vrcholů



Výběr vrcholů je naprogramován v souboru *filterModalWindow.js*, kde bylo upraveno původní filtrování vrcholů, které pouze vrcholy zvýraznilo. Nyní jsou vrcholy, které neodpovídají kritériu při výběru odstraňovány z grafu (jsou schovány). Odstraněné vrcholy jsou uloženy v polích v hlavním souboru aplikace pro zobrazování grafu *showGraphApp.js*. Při resetování výběru jsou vrcholy do grafu opět vráceny. Výběr vrcholů je v aplikaci reprezentován v navigační liště tlačítkem *Select nodes*. Kritéria se skládají za sebe, tudíž po aplikování prvního kritéria je další aplikováno již pouze na ty vrcholy, které zůstaly v grafu po aplikaci prvního.

## Závěr

Žádná ze změn nemá vliv na základní strukturu architektury. Změny byly spíše aditivní.