

# Towards Understanding and Fixing Upstream Merge Induced Conflicts in Divergent Forks: An Industrial Case Study



Chungha Sung\*, Shuvendu K. Lahiri<sup>†</sup>, Mike Kaufman<sup>†</sup>,  
Pallavi Choudhury<sup>†</sup>, Jessica Wolk<sup>†</sup>, Chao Wang\*  
University of Southern California\* & Microsoft Corporation<sup>†</sup>



## Abstract

*Divergent forks* are a common practice in open-source software development to perform long-term, independent and diverging development on top of a popular source repository. However, keeping such divergent downstream forks in sync with the upstream source evolution poses engineering challenges in terms of frequent merge conflicts. In this work, we conduct the first industrial case study of frequent merges from upstream and the resulting merge conflicts, in the context of Microsoft Edge development. The study consists of two parts. First, we describe the nature of merge conflicts that arise due to merges from upstream. Second, we investigate the feasibility of automatically fixing a class of merge conflicts related to build breaks that consume a significant amount of developer time to root-cause and fix. Towards this end, we have implemented a tool MrgBldBrkFixer and evaluate it on three months of real Microsoft Edge development data, and report encouraging results

## Study of merge conflicts in divergent forks

Conflict	Details	# of commits	%
Textual	Textual-level conflict resolution	1183	53.3
Build Break	Fixes of ill-formed files	158	36.7
	- Stylelint fixes	(31)	
	- Parse error fixes	(19)	
	- Uncategorized	(108)	
	Build script file fixes (js, .gn, .json, etc.)	259	
	Structural fixes in C++ files	398	
Test Failure	Macro fixes in unit test	169	9.9
	Flag file fixes	51	
Total		2218	

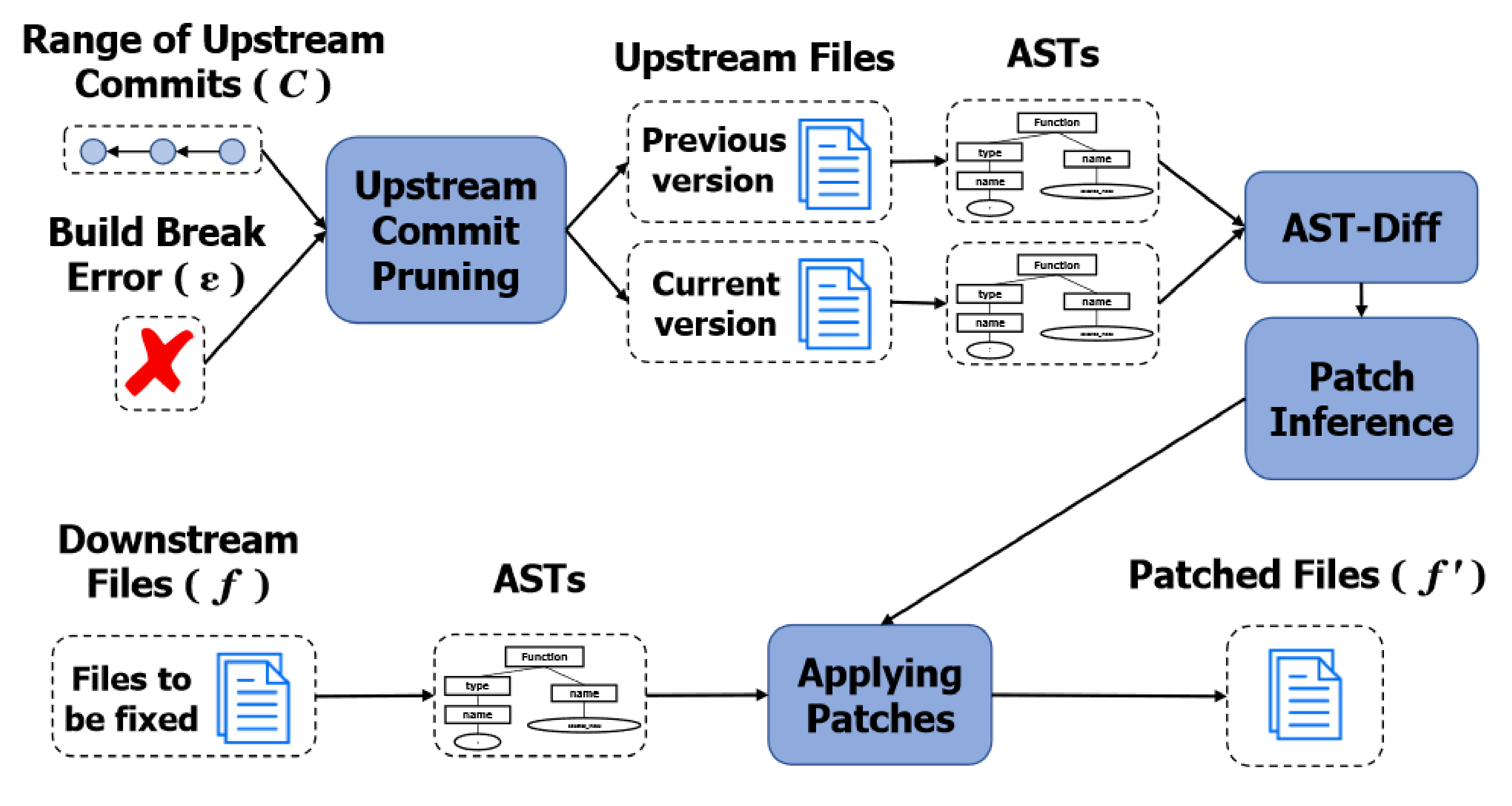
### Our target

- Include Statement Update
- Entire Function Definition/Call Update
- Function Name Update
- Function Type/Specifier Update
- Function Parameter/Argument Update
- Function Parameter/Argument's Type Update
- Class/Namespace/Enum Reference Update
- Uncategorized

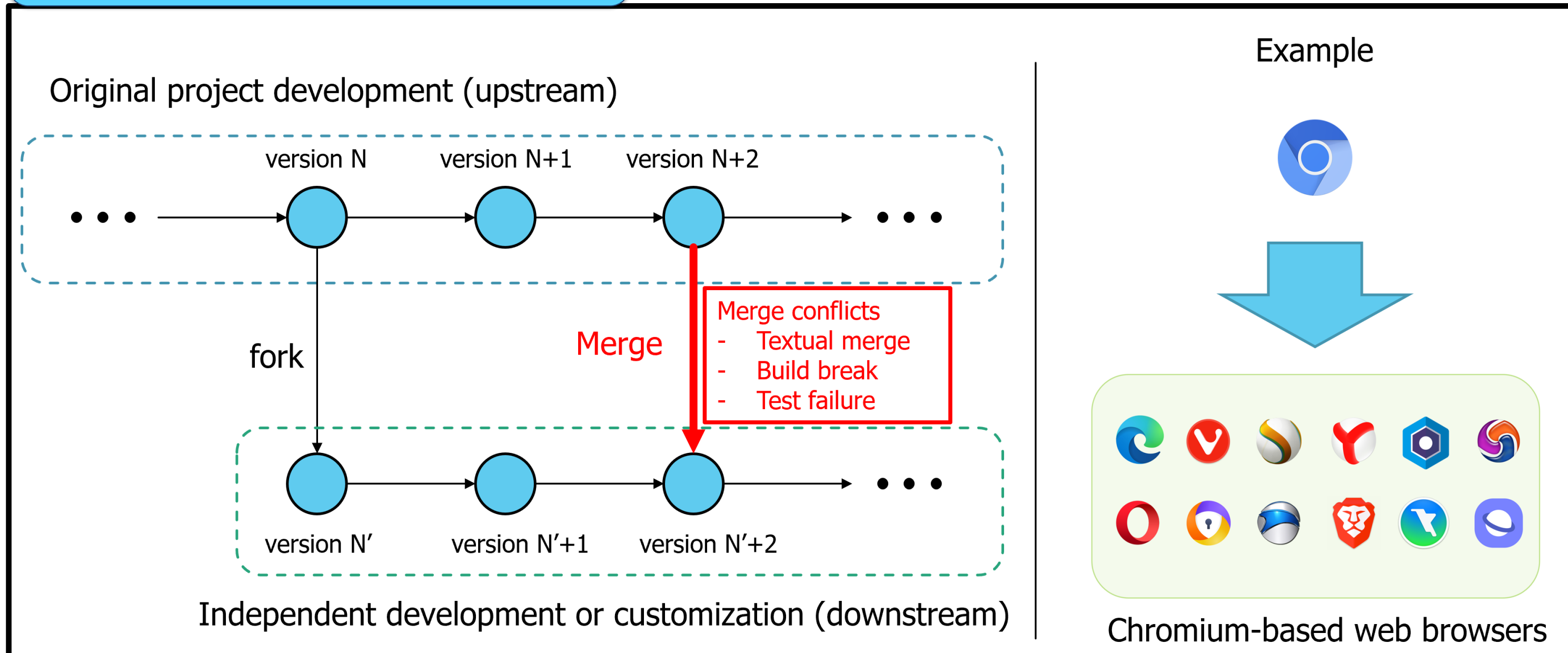
"For each upstream induced build break, it takes at least 30 minutes to hours for developers to resolve. The main burden for the developers is they need to look up the history of upstream changes."

- A senior manager in the Microsoft Edge development team

## Overview of automatic patch for build breaks



## Divergent forks?



## Problems of merges from upstream

**Build break file in Edge**

```

164
165 void EdgeTranslateBubbleView::HandleComboBoxPerformanceAction(
166     ComboBoxID sender_id) {
167     views::ComboBox* combobox = GetLanguageComboBox();
168     if (model_>GetTargetLanguageIndex() == combobox->GetSelectedIndex())
169         return;
170     model_>UpdateTargetLanguageIndex(combobox->GetSelectedIndex());
171     // If the model state is VIEW_STATE_AFTER_TRANSLATE
172
173
                
```

Undefined symbol

Edge code fix (05/06/2019, 7ab5c10)

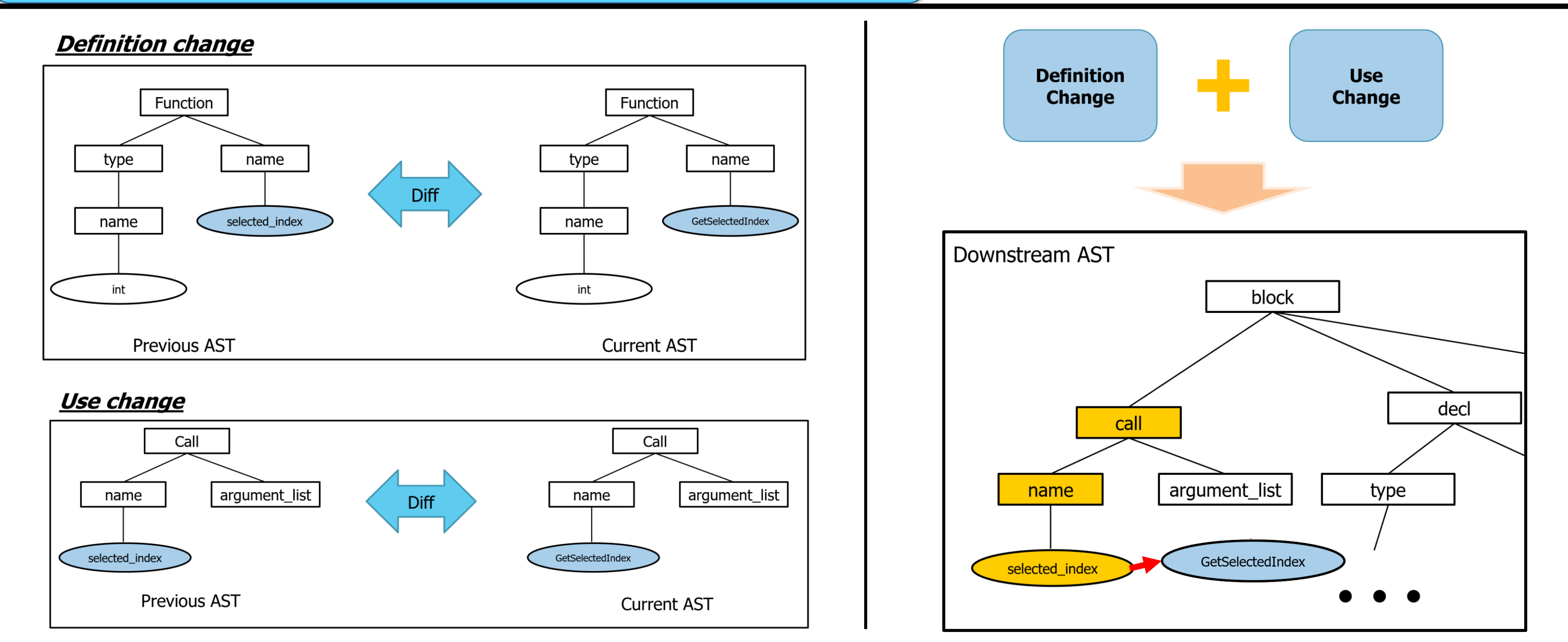
**API def. change in Chromium**

```

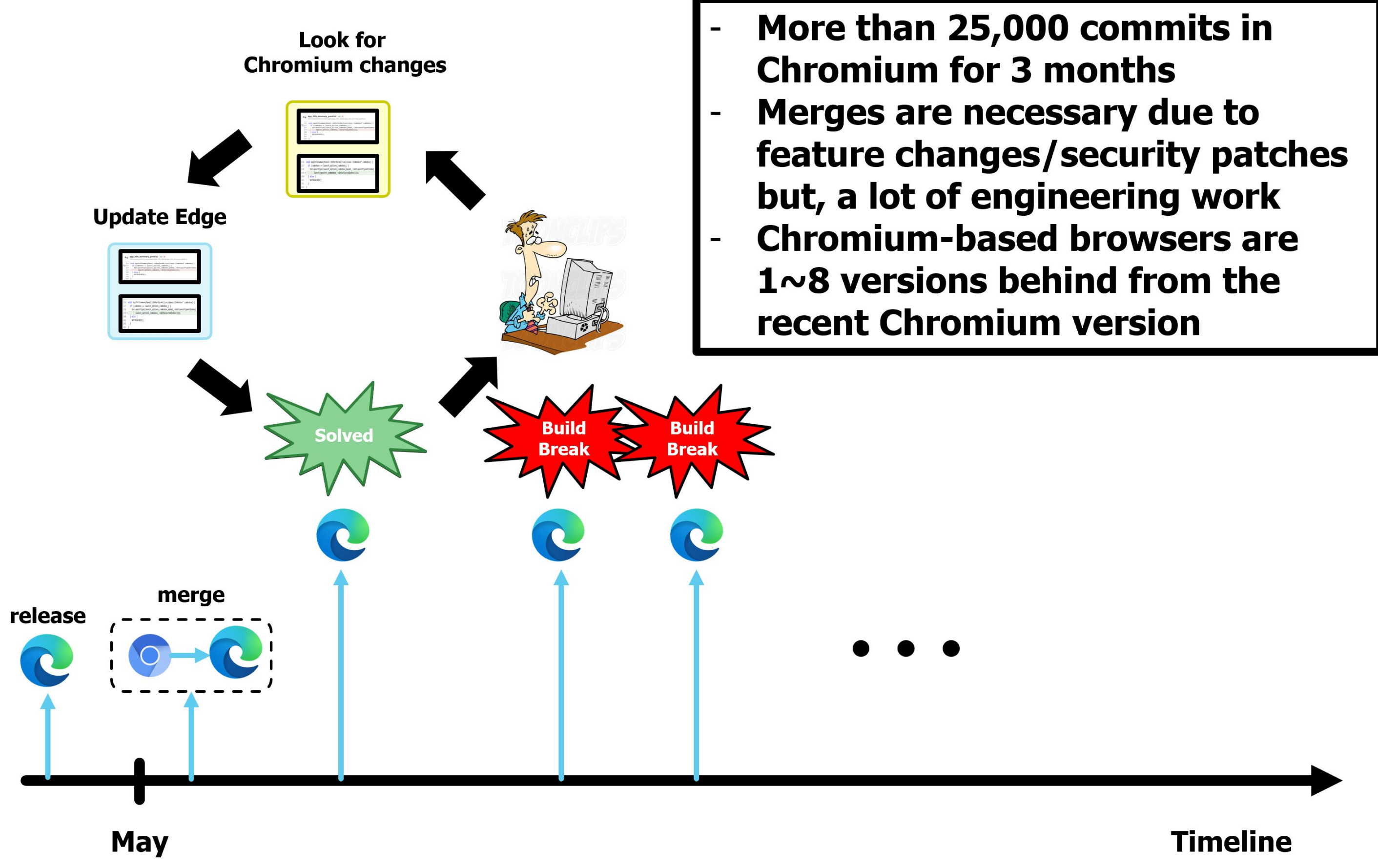
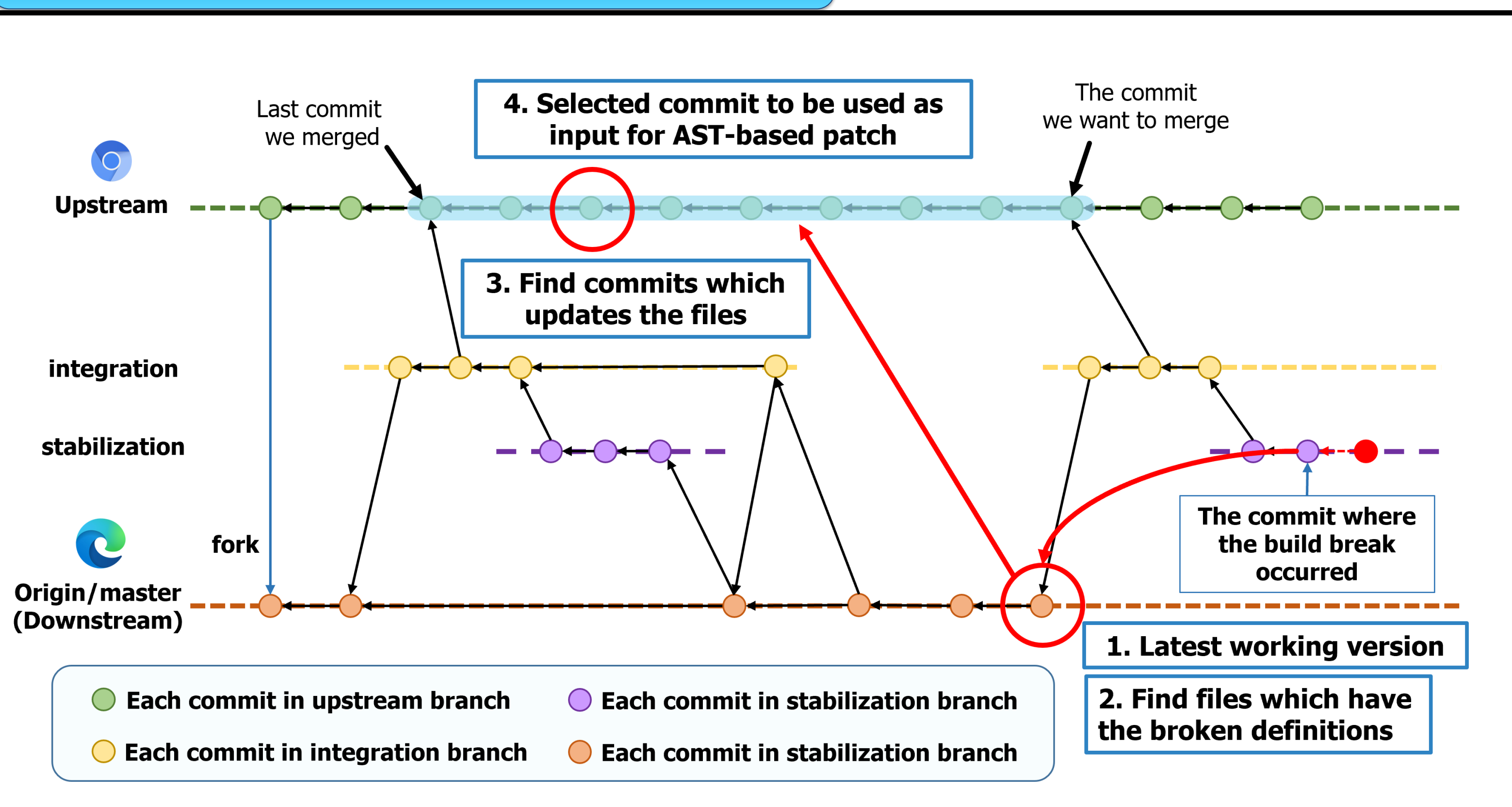
60 // Informs the combobox that its model changed.
61 void ModelChanged();
62
63 // Gets/Sets the selected index.
64 int GetSelectedIndex() const { return selected_index_; }
65 void SetSelectedIndex(int index);
66
                
```

Chromium code update (05/02/2019, 0b079bf5)

## AST-based patch inference



## Upstream commit pruning



More details can be found on our paper

"Towards Understanding and Fixing Upstream Merge Induced Conflicts in Divergent Forks: An Industrial Case Study", ICSE-SEIP