On the Interaction of Feature Toggles

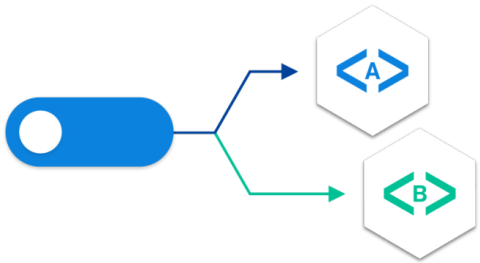
Xhevahire Tërnava, Luc Lesoil, Georges Aaron Randrianaina,
Djamel Eddine Khelladi, and Mathieu Acher

Université de Rennes 1, Inria/IRISA, CNRS, IUF
Rennes, France

VaMoS'22 - February 24, 2022

Context

- A feature toggle is a binary condition that controls whether a feature appears in the system or not




```
function reticulateSplines() {

    var useNewAlgorithm = false;
    // UNCOMMENT IF YOU ARE WORKING ON THE NEW SR ALGORITHM
    // useNewAlgorithm = true;

    if (useNewAlgorithm) {
        return enhancedSplineReticulation();
    } else {
        return oldFashionedSplineReticulation();
    }
    //The rest of code is omitted
}
```

Feature Toggle

Feature Toggle Point

- They enable trunk-based development (Google, Microsoft, Facebook)
- They are used for continuous deployment
- Mostly, they are *temporary* in the system (vs. configuration options)

Motivation

- The adoptation of trunk-based development is growing
- Feature toggles are perceived as a simple and light technique

Listing 1 The defined toggles: kops/pkg/featureflag/featureflag.go

```
1  var (   //The rest of 21 feature toggles are omitted
2          TerraformJSON = new("TerraformJSON", Bool(false))
3          TerraformManagedFiles = new("TerraformManagedFiles", Bool(true))
4  ) // The rest of code is omitted
```

Listing 2 Usage: kops/upup/pkg/fi/cloudup/terraform/target.go

```
1  func (t *TerraformTarget) Finish(taskMap map[string]fi.Task) error {
2          //The rest of code is omitted
3          if featureflag.TerraformJSON.Enabled() {
4                  if featureflag.TerraformManagedFiles.Enabled() {
5                          return errors.New("TerraformJSON cannot be
6                          used with TerraformManagedFiles")
7                  }
8                  err = t.finishJSON()
9          } else {
10                 err = t.finishHCL2()
11         }
12         return nil
13 }
```

- Are they simple to deal with in real projects, especially over time?
- What is their interaction complexity within a codebase?

Study Design

– Goal and Research Questions –

Goal: To explore the presence of feature toggles interactions in a system from the developer's viewpoint

RQ_1: Do feature toggles interact with each other in order to enable some system functionalities (*i.e.*, features)?

RQ_2: Do feature toggles and their interactions have a tendency to be multiplied over time?

Study Design

– Goal and Research Questions –

Goal: To explore the presence of feature toggles interactions in a system from the developer's viewpoint

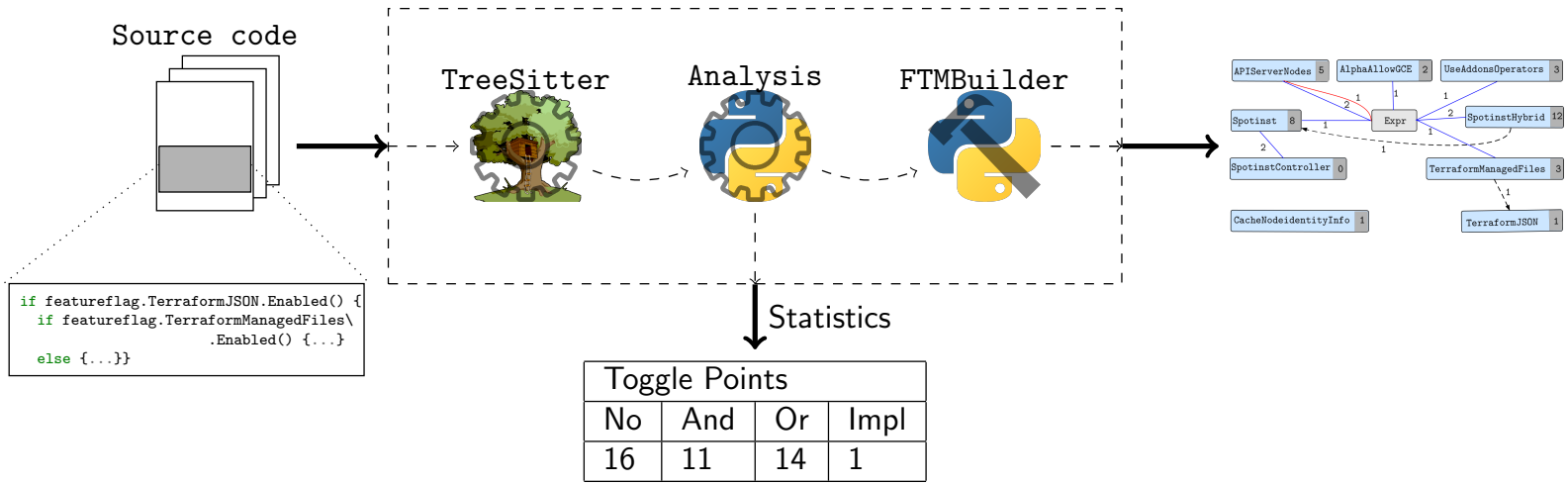
RQ_1: Do feature toggles interact with each other in order to enable some system functionalities (*i.e.*, features)?

RQ_2: Do feature toggles and their interactions have a tendency to be multiplied over time?

- We analysed 5 Go open-source projects
  - Boulder : 0.52M LoC, *** 4k
  - Juju : 11.12M LoC, *** 1,9k
  - Kops : 2.61M LoC, *** 13,4k
  - Kubernetes : 4.58M LoC, *** 18,2k
  - Loomchain : 0.10M LoC, *** 136

Study Design

– Our Approach –



Statistics

Toggle Points			
No	And	Or	Impl
16	11	14	1

- For RQ_1: Feature toggles interactions in 1 recent release
- For RQ_2:
  - Feature toggles changes throughout the project history
  - Feature toggles interactions in the 1st release for last 5 years

Results of RQ_1 (1/4)

Do feature toggles interact with each other in order to enable some system functionalities?

- The overall number of feature toggles that interact

Software System	# Feature Toggles			
	Unused	Independent	Interacting	Total
Boulder	5	6	6	17
Juju	1	9	3	13
Kops	1	11	11	23
Kubernetes	31	24	53	108
Loomchain	21	21	13	55
In percentage	27%	33%	40%	

Results of RQ_1 (2/4)

Do feature toggles interact with each other in order to enable some system functionalities?

- The kind of feature toggles interactions

Software System	# Feature Toggle Points				
	No	AND	OR	IMPLIES	Total
Boulder	13	16	0	1	23
Juju	16	11	14	1	37
Kops	58	10	4	13	69
Kubernetes	170	130	8	24	278
Loomchain	91	16	08	10	109
In percentage	67%	69%	13%	18%	

Results of RQ_1 (3/4)

Do feature toggles interact with each other in order to enable some system functionalities?

- The scattering of feature toggles in files

Software System	# Files			
	With Feature Toggles	Range	Total	In percentage
Boulder	16	0 - 2	2,332	0.69%
Juju	49	0 - 12	6,602	0.74%
Kops	29	0 - 11	11,078	0.26%
Kubernetes	297	0 - 23	17,503	1.70%
Loomchain	39	0 - 7	472	8.26%

Results of RQ_1 (4/4)

Do feature toggles interact with each other in order to enable some system functionalities?

- Interactions (40%):
  - 7% between feature toggles
  - 33% between feature toggles and code expressions

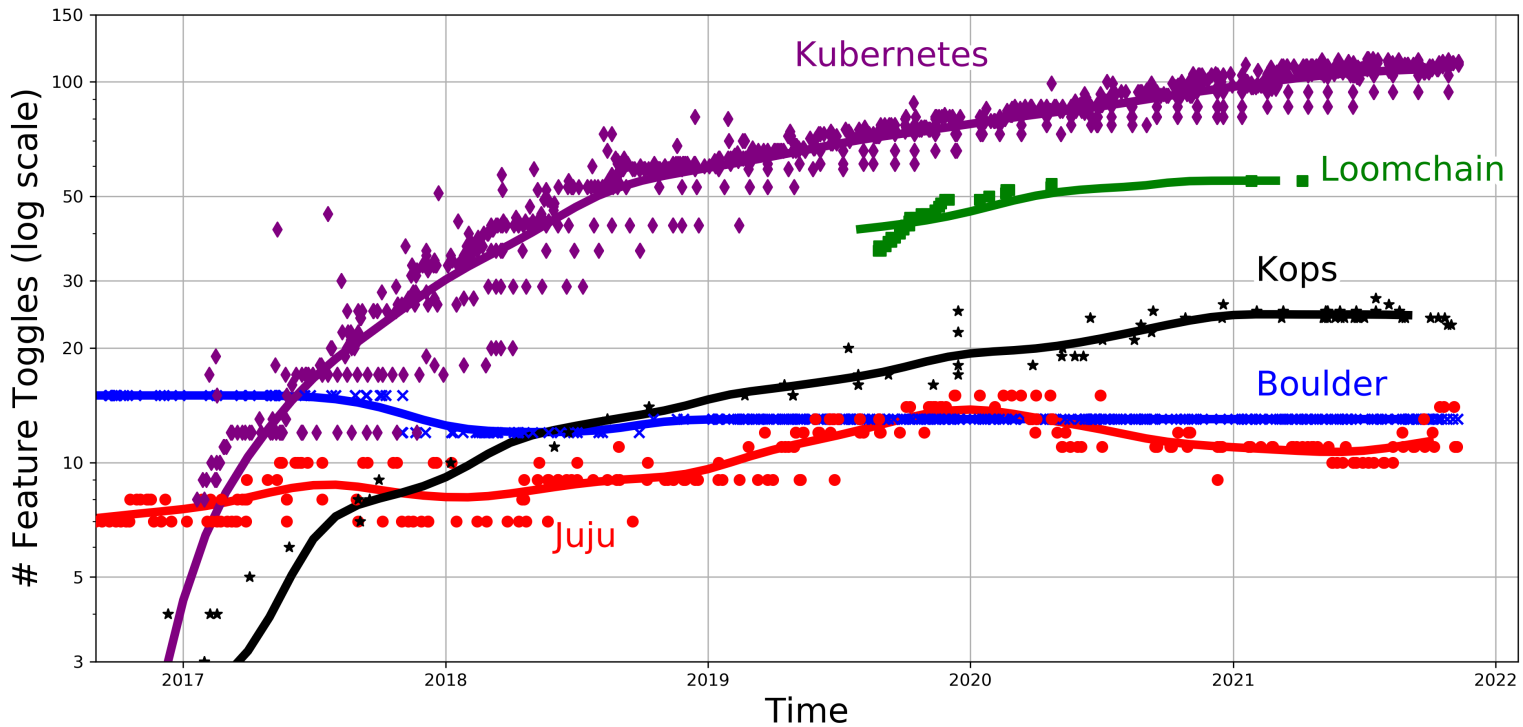
```
1778            if features.Enabled(features.StoreRevokerInfo) && revokedBy != 0 {
1779                    req.RevokedBy = revokedBy
1780            }
```

- Up to 2 feature toggles are involved in an interaction
- 27% of feature toggles are unused
- Feature toggles are concentrated in only 1.13% of files

Results of RQ_2 (1/3)

Do feature toggles and their interactions have a tendency to be multiplied over time?

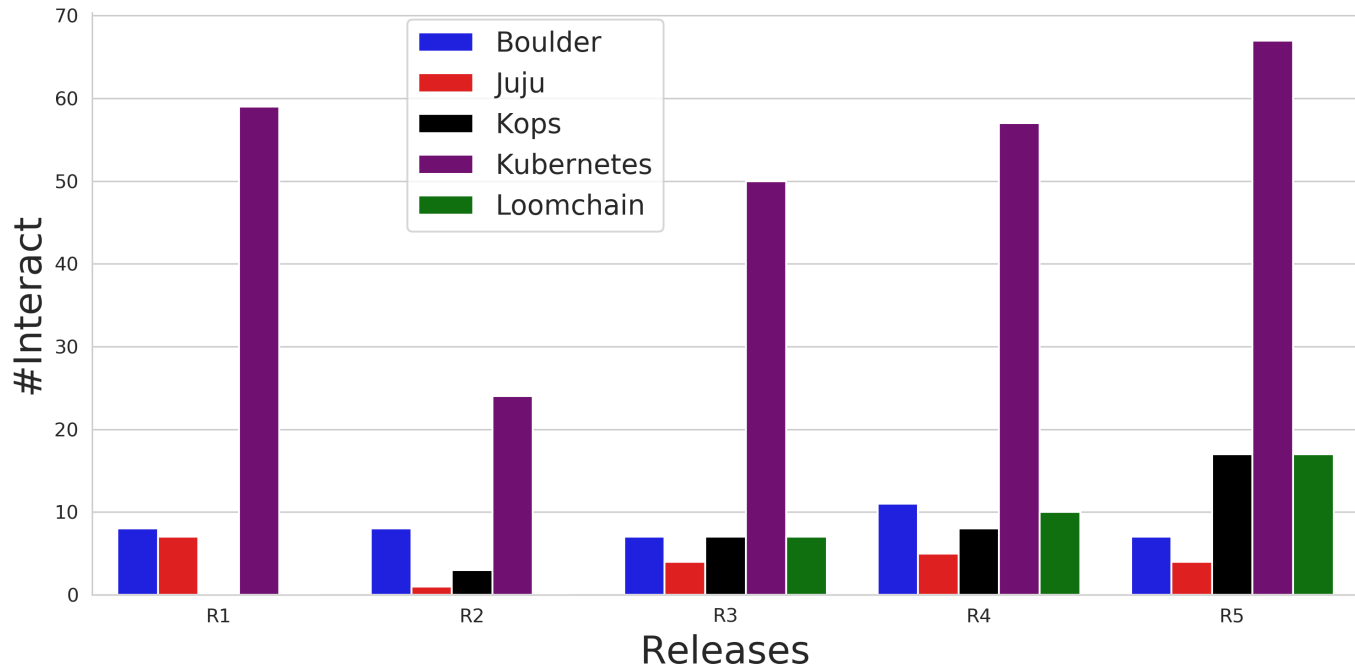
- The multiplication of #feature toggles over the time



Results of RQ_2 (2/3)

Do feature toggles and their interactions have a tendency to be multiplied over time?

- The multiplication of feature toggles interactions over the time



Results of RQ_2 (3/3)

Do feature toggles and their interactions have a tendency to be multiplied over time?

- Feature toggles interactions tend to multiply over time by 22%

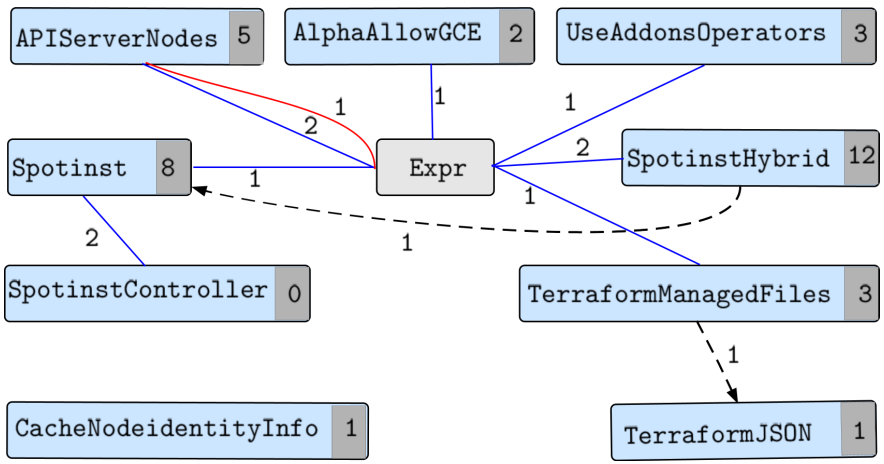
  ↓ Boulder (18%) and Juju (6%)
  ↑ Kops (65%), Kubernetes (29%), and Loomchain (39%)

- Correlation between the #feature toggles and their interactions

More: Kops (0.90), Loomchain (0.88), and Kubernetes (0.39)
Less: Juju (0.04) and Boulder (-0.02)

An extracted Feature Toggle Model

- To know how and which feature toggles interact in a system, we propose to model them into a Feature Toggle Model (FTM)



An excerpt from the FTM of Kops (9 from its 23 feature toggles)

Summary: On the Interaction of Feature Toggles

Contributions:

- A first empirical evaluation of interaction of feature toggles
  - 7% of feature toggles interact with each other
  - Their interactions tend to multiply over time for 22%
  - Their interactions can be correlated with their number (3/5)
- The proposition of a Feature Toggle Model (FTM)
- The availability of the mining approach and artifacts: https://github.com/llesoil/poc_ftm

Limitations:

- We mined only the *if-else* statements
- Feature toggle can be assigned to a variable and then to be evaluated
- The EXCLUDES interactions of feature toggles are not identified