KUBERNETES COMMUNITY DAYS BEIJING 2025

即刻受益：从自动化到智能化，代码扫描工具升级实践

Upgrade Bot to AI Agent,

benefit Open Source Communities

CNCF slack:@SamYuan | Github ID:SamYuan1990

Part 01

灵感

REACT : SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS

即刻受益：从自动化到智能化，代码扫描工具升级实践

- 通过可靠的第三方数据从而提高LLM对于问题的理解，提高正确率。论文中使用了Wiki作为第三方外部数据源。

	Type	Definition	ReAct	CoT
Success	True positive	Correct reasoning trace and facts	94%	86%
	False positive	Hallucinated reasoning trace or facts	6%	14%
Failure	Reasoning error	Wrong reasoning trace (including failing to recover from repetitive steps)	47%	16%
	Search result error	Search return empty or does not contain useful information	23%	-
	Hallucination	Hallucinated reasoning trace or facts	0%	56%
	Label ambiguity	Right prediction but did not match the label precisely	29%	28%

Table 2: Types of success and failure modes of ReAct and CoT on HotpotQA, as well as their percentages in randomly selected examples studied by human.

Part 02

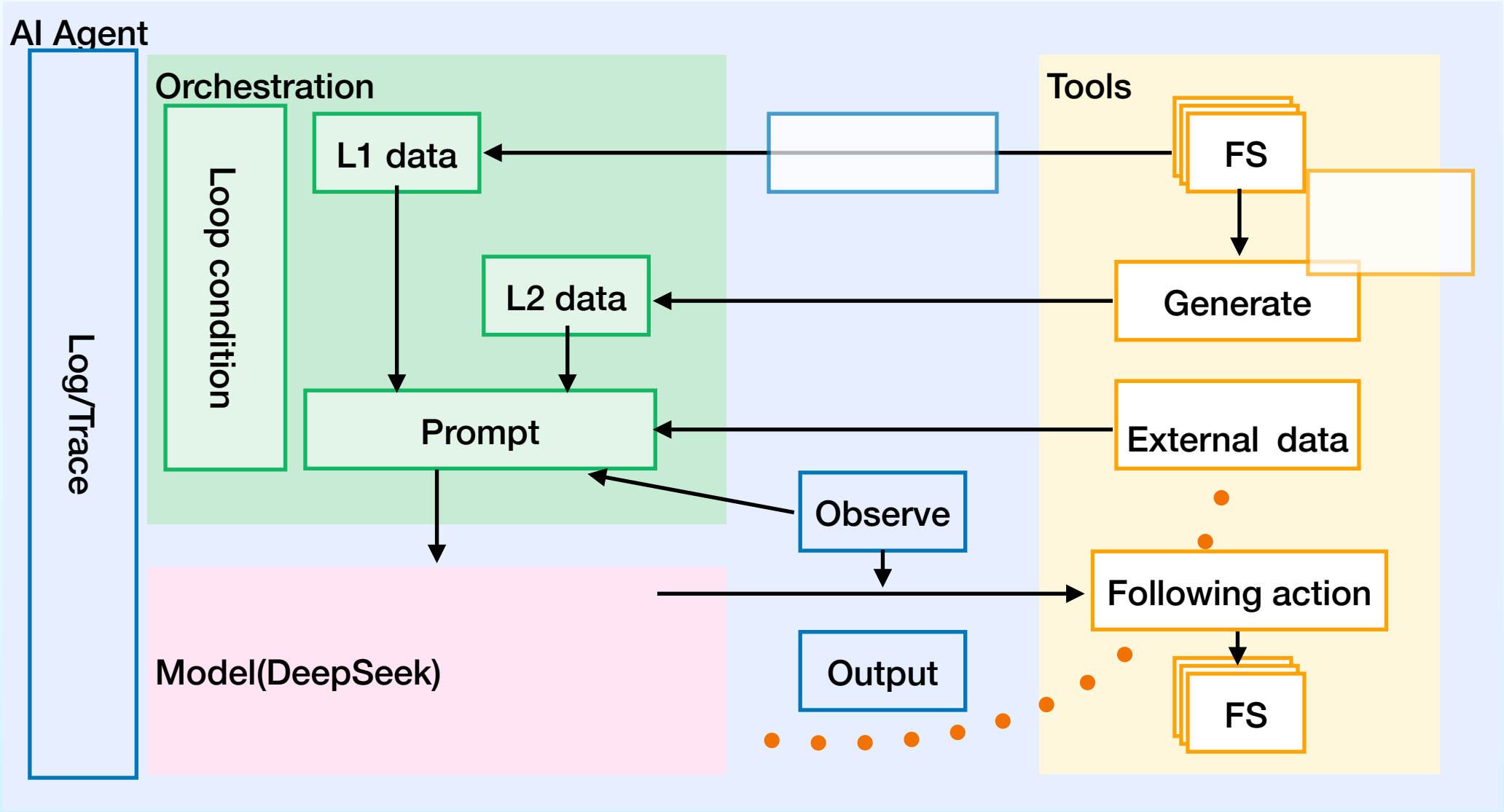
设计，实现

在pipeline中，我们去哪里找寻事实?

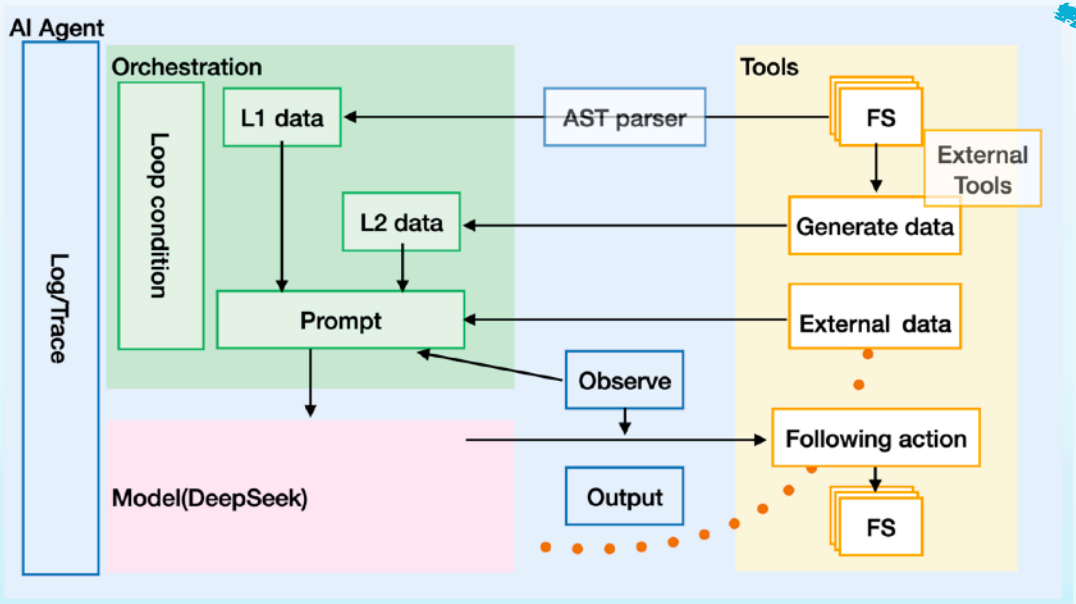
即刻受益：从自动化到智能化，代码扫描工具升级实践

场景	工具	目的	内容	措施
单元测试	JUnit，Go test	验证代码功能健壮性	测试通过率，代码覆盖率	修复测试用例，补全测试覆盖率
静态代码分析	ESLint	检查代码风格，潜在缺陷	代码异味，潜在问题	重构代码，安全提升
依赖扫描	Snyk	检测第三方已知漏洞	CVE编号，风险	升级代码版本，采取安全措施
安全扫描	OWASP	检测代码漏洞	漏洞类型，风险等级	修复漏洞代码，提高软件安全

即刻受益：从自动化到智能化，代码扫描工具升级实践



即刻受益：从自动化到智能化，代码扫描工具升级实践



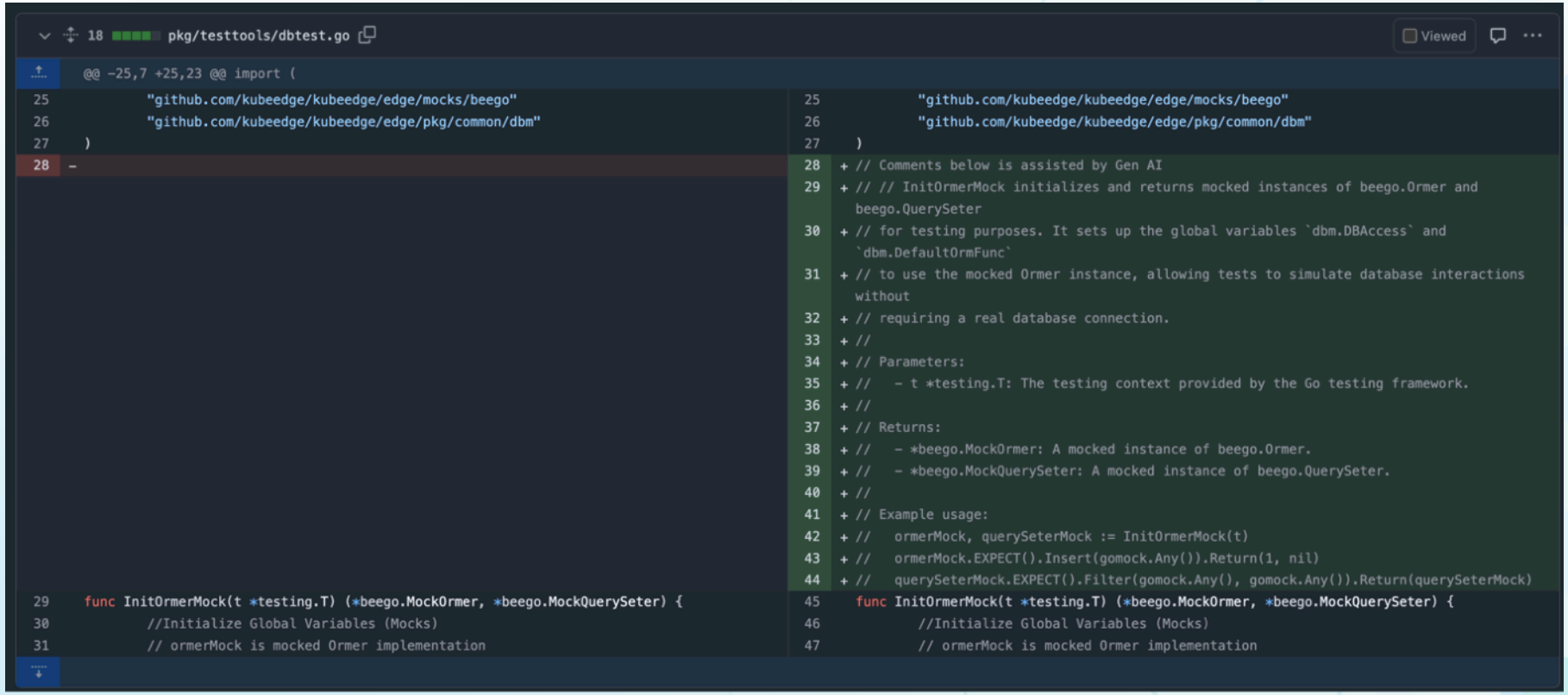

```yaml
        repository: SamYuan1990/kubeedge
    - uses: actions/setup-go@v5
    - name: generate CVE.json
      run: |
        curl -sSfL https://raw.githubusercontent.com/anchore/syft/main/install.sh | sh -s -- -b ./
        go install github.com/devops-kung-fu/bomber@latest
        ./syft scan kubeedge/${{matrix.IMAGE}}:v1.20.0 -o cyclonedx-json -vv  > sbom.json
        cat sbom.json
        bomber scan ./sbom.json --output=json --debug > cve.json
        cat cve.json
    - name: use this action to generate suggestion for deployment settings
      id: Lint_with_LLM
      uses: SamYuan1990/OpenAI_CodeAgent-action@main
      with:
        baseURL: https://api.deepseek.com
        apiKey: ${{ secrets.API_KEY }}
        model: deepseek-chat
        dirpath: '/workdir'
        deploymentfile: /workdir/${{ matrix.PATH }}
        runType: CVE2Deployment
    - name: output check
      shell: bash
      run: |
        echo '${{ steps.Lint_with_LLM.outputs.avg_prompt_precent }}'
        echo '${{ steps.Lint_with_LLM.outputs.avg_content_precent }}'
        echo '${{ steps.Lint_with_LLM.outputs.avg_time_usage }}'
        echo '${{ steps.Lint_with_LLM.outputs.avg_inputToken }}'
        echo '${{ steps.Lint_with_LLM.outputs.avg_outputToken }}'

    - name: Create new issue
      uses: imjohnbo/issue-bot@v3
      if: ${{ inputs.dryrun == false }}
      with:
        title: CVE cross check with deployment on container image ${{matrix.IMAGE}}
        body: |-
          :wave: Hi maintainers, here is LLM's deployment suggested according to your CVSS scan result.
          ${{ steps.Lint_with_LLM.outputs.LLMresponse }}
```

PR push back as Bot

Part 03

效果，指标

即刻受益：从自动化到智能化，代码扫描工具升级实践



18 pkg/testtools/dbtest.go

Viewed

```
@@ -25,7 +25,23 @@ import (
        "github.com/kubeedge/kubeedge/edge/mocks/beego"
        "github.com/kubeedge/kubeedge/edge/pkg/common/dbm"
)
-
+// Comments below is assisted by Gen AI
+// // InitOrmerMock initializes and returns mocked instances of beego.Ormer and beego.QuerySeter
+// for testing purposes. It sets up the global variables `dbm.DBAccess` and `dbm.DefaultOrmFunc`
+// to use the mocked Ormer instance, allowing tests to simulate database interactions without
+// requiring a real database connection.
+//
+// Parameters:
+//   - t *testing.T: The testing context provided by the Go testing framework.
+//
+// Returns:
+//   - *beego.MockOrmer: A mocked instance of beego.Ormer.
+//   - *beego.MockQuerySeter: A mocked instance of beego.QuerySeter.
+//
+// Example usage:
+//   ormerMock, querySeterMock := InitOrmerMock(t)
+//   ormerMock.EXPECT().Insert(gomock.Any()).Return(1, nil)
+//   querySeterMock.EXPECT().Filter(gomock.Any(), gomock.Any()).Return(querySeterMock)
 func InitOrmerMock(t *testing.T) (*beego.MockOrmer, *beego.MockQuerySeter) {
        //Initialize Global Variables (Mocks)
        // ormerMock is mocked Ormer implementation
```

即刻受益：从自动化到智能化，代码扫描工具升级实践

KCD GROWING CLOUD NATIVE TOGETHER BEIJING

2. Use Non-Root User

- Run the container as a non-root user to minimize the impact of potential exploits.
- Add the following to your container spec:

```
securityContext:
 runAsNonRoot: true
 runAsUser: 1000
 runAsGroup: 2000
```

3. Enable Read-Only Root Filesystem - Since the container does not need to write to the filesystem, set the root filesystem to read-only. - Add the following to your container spec:

```
securityContext:        readOnlyRootFilesystem: true
```

4. Drop Unnecessary Capabilities

- Drop all Linux capabilities that are not required by the container.
- Add the following to your container spec:

```
securityContext:
  capabilities:
    drop:
      - ALL
```

即刻受益：从自动化到智能化，代码扫描工具升级实践

Explanation:

- **Original Code:**

```
const std::string& AdaptiveMaxConcurrency::CONSTANT() {
    static std::string* s = new std::string("constant");
    return *s;
}
```

  - The original code uses a `static std::string*` which leads to a memory leak since the allocated memory is never freed.

- **Fixed Code:**

```
const std::string& AdaptiveMaxConcurrency::CONSTANT() {
    static std::string s("constant");
    return s;
}
```

  - The fixed code uses a `static std::string` instead of a pointer, which ensures that the memory is managed automatically by the C++ runtime, eliminating the risk of memory leaks and simplifying the code.

即刻受益：从自动化到智能化，代码扫描工具升级实践

指标	解释	例子
提示词占比	提示词占任务的比重	请帮我为这个函数写一段描述, func add(…
内容占比	具体内容占任务的比重	请帮我为这个函数写一段描述, func add(…
输出Token数量	length(output)	*// func add is a sum function which…*
运行时长	大模型执行单次任务的时间	

即刻受益：从自动化到智能化，代码扫描工具升级实践

指标\功能	文档生成	部署建议	代码脆弱扫描
提示词占比	16.029%	5.685714285714 29%	54.24%
内容占比	83.622%	93.71428571428 57%	45.65%
输出Token数量	430.3	1207.7142857142 9	742
运行时长(s)	26.022	61.631714285714 3	43.89

Thanks

CNCF slack:@SamYuan | Github ID:SamYuan1990