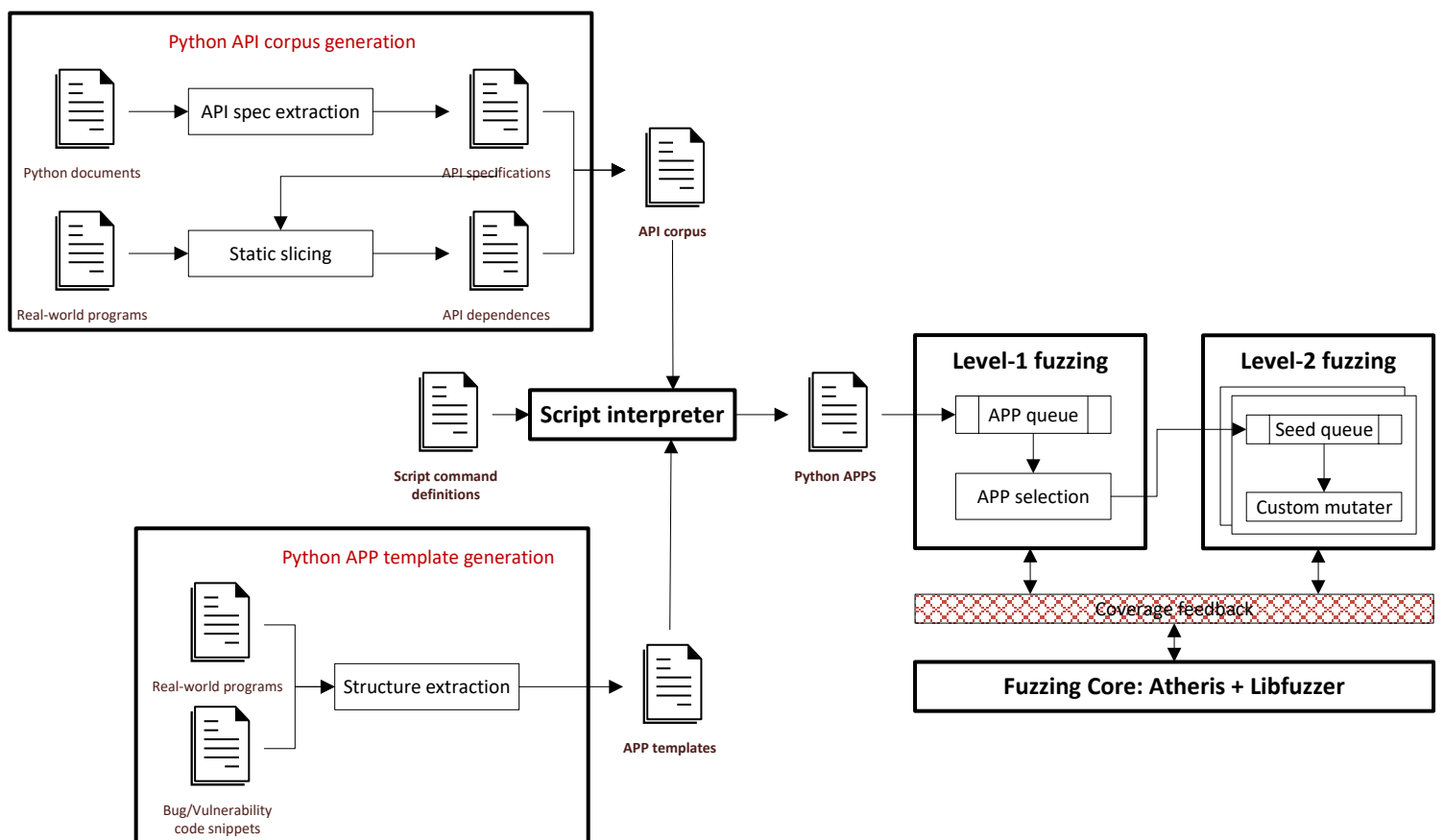
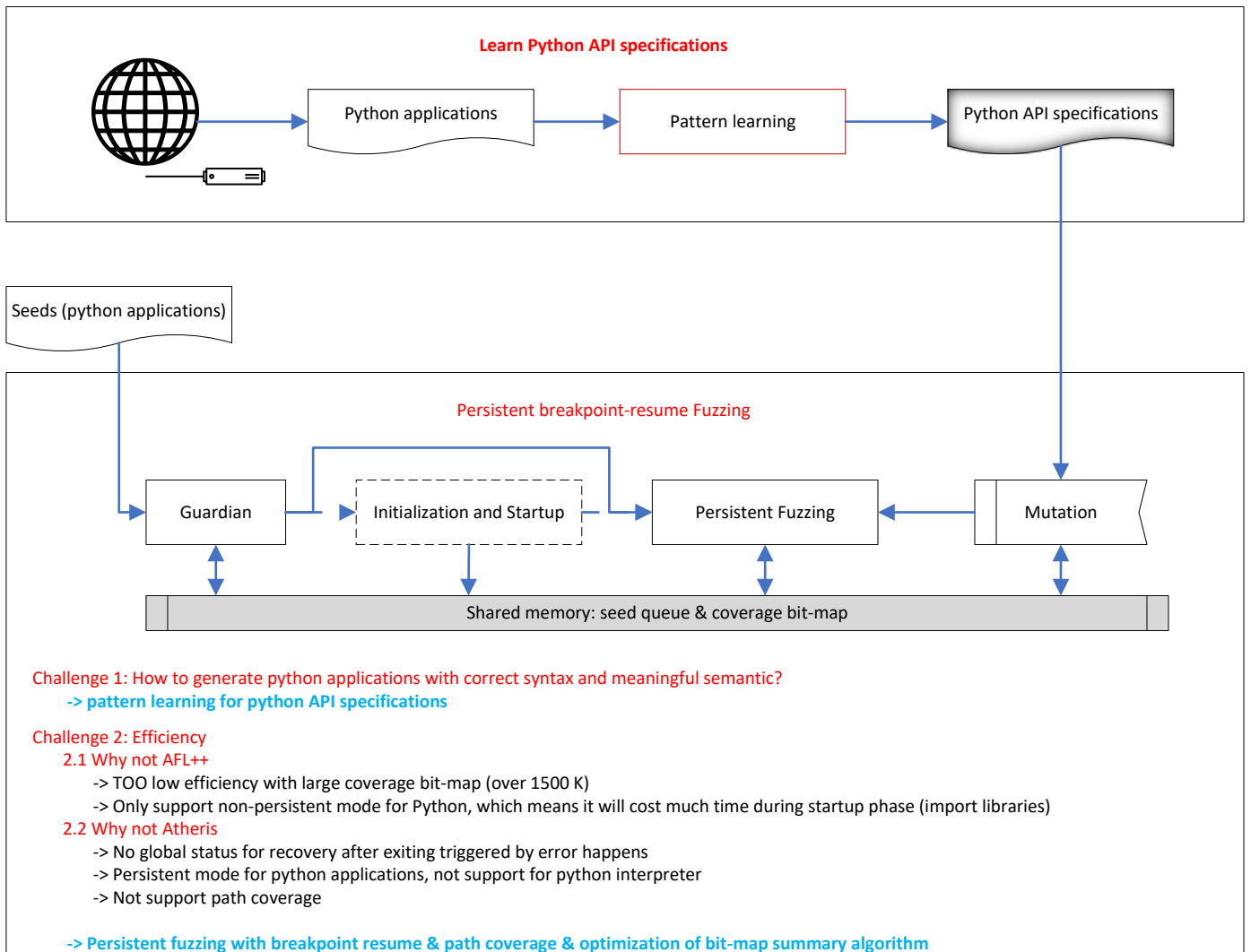
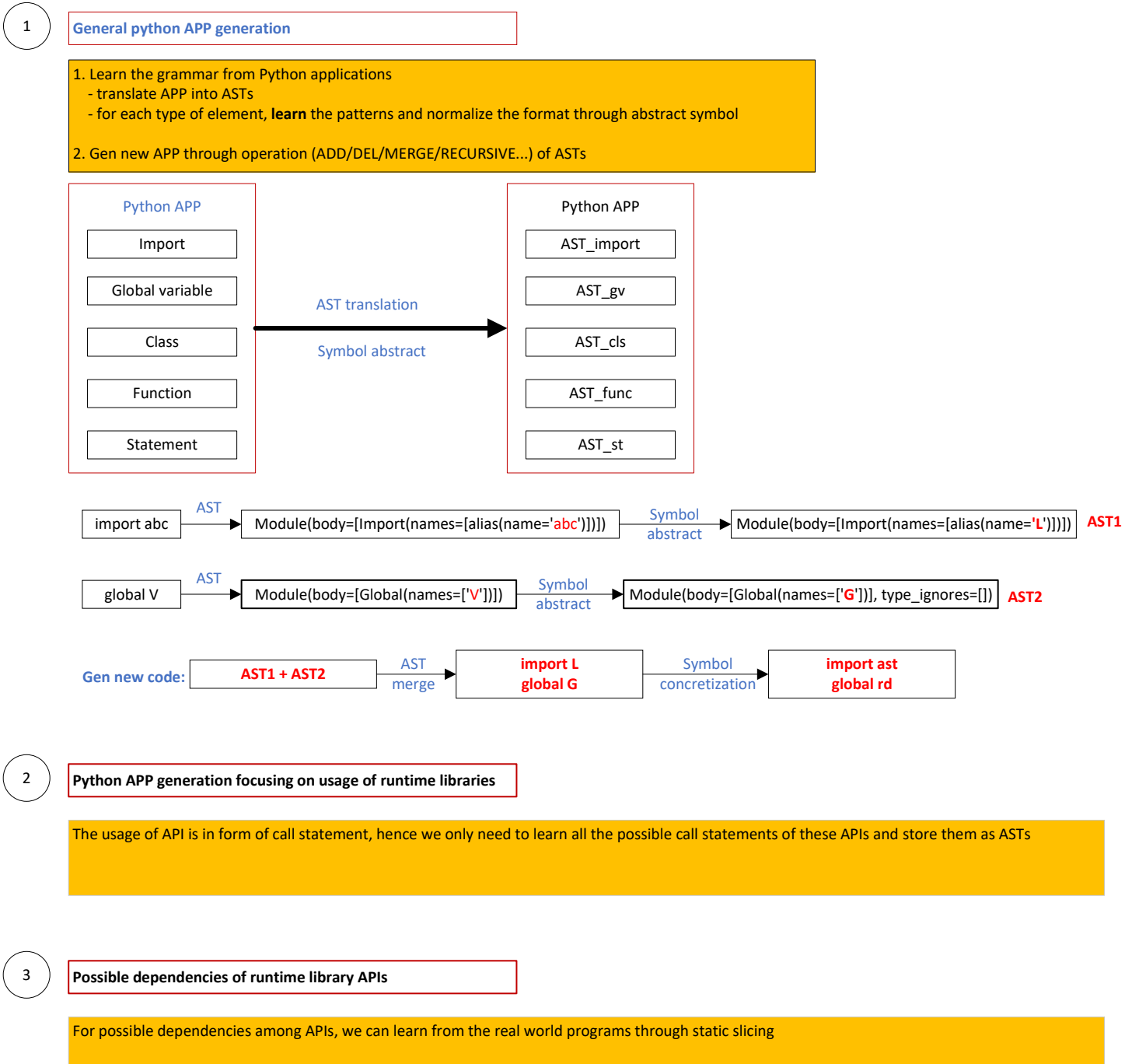


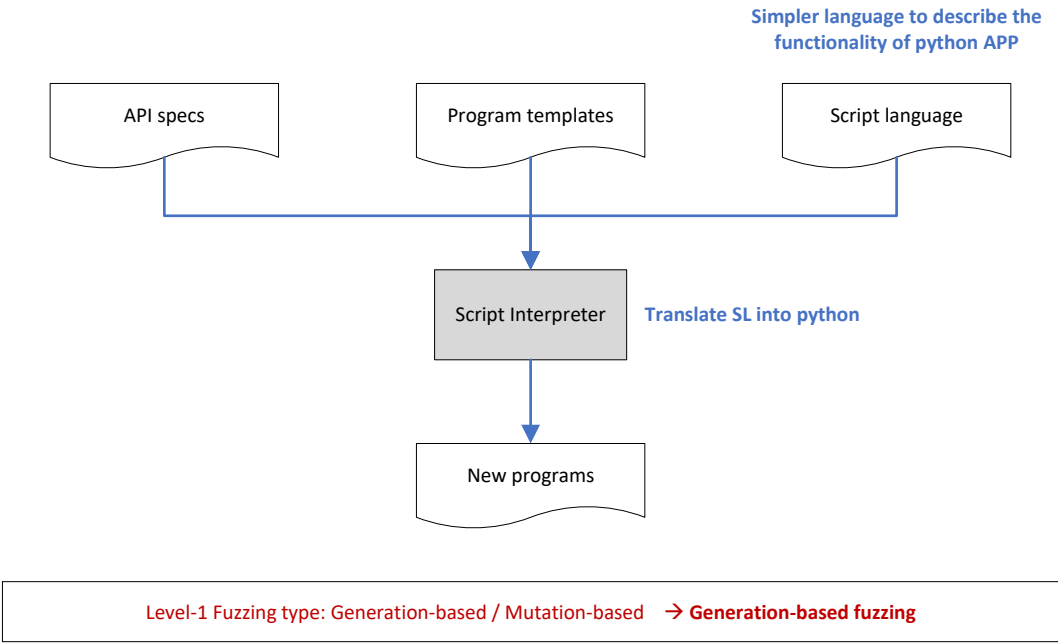
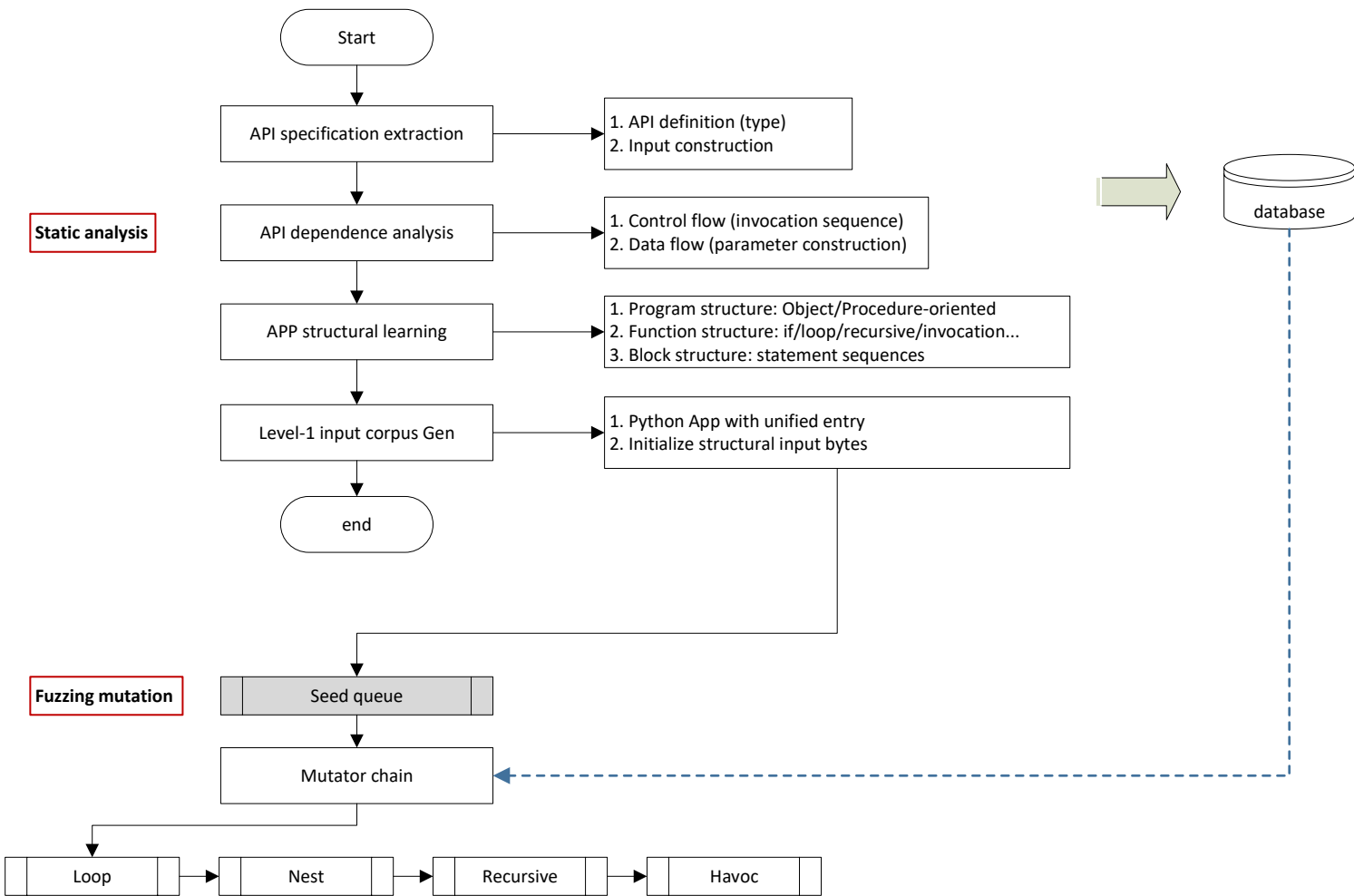
Overview



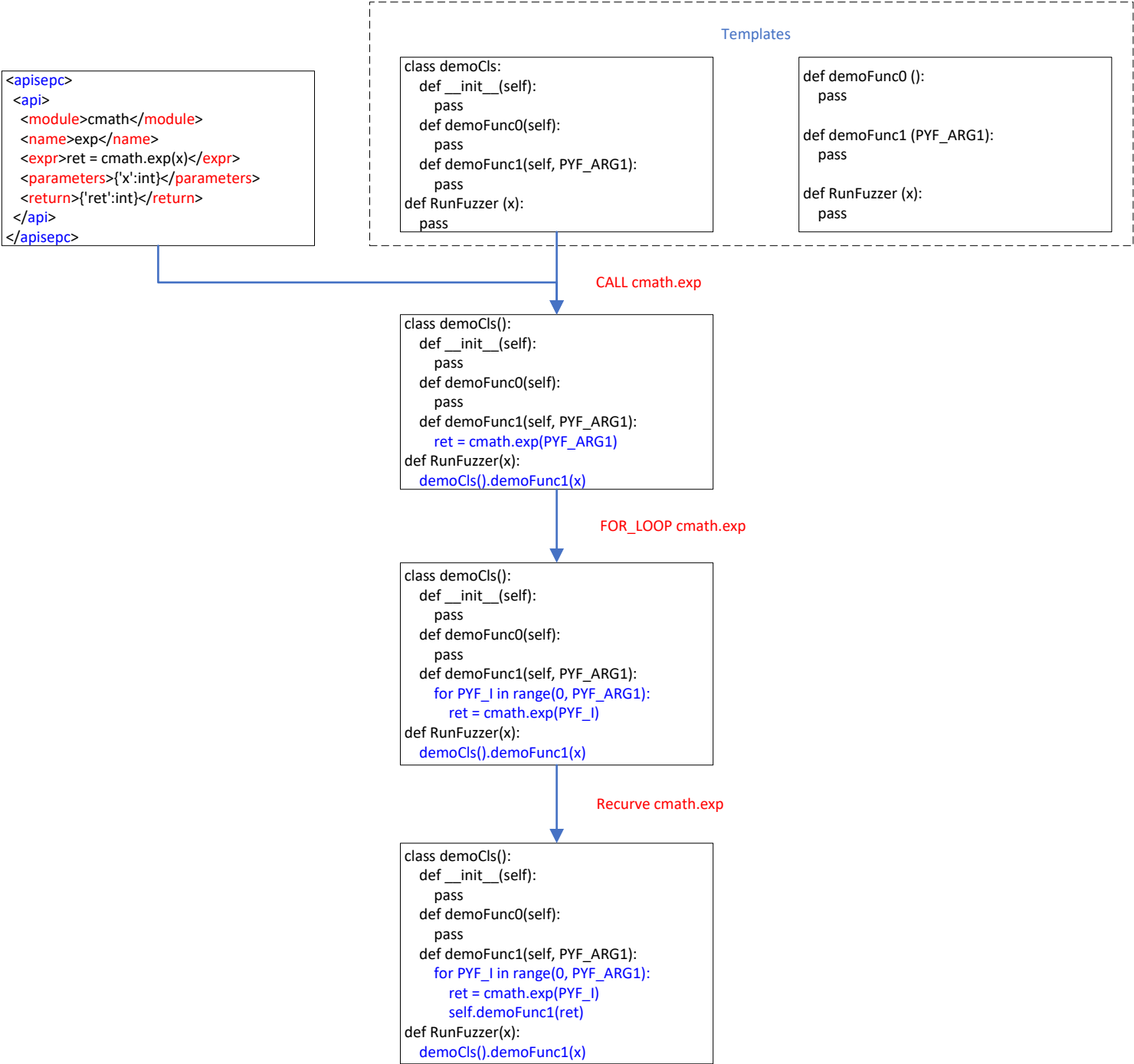
Python APP generation



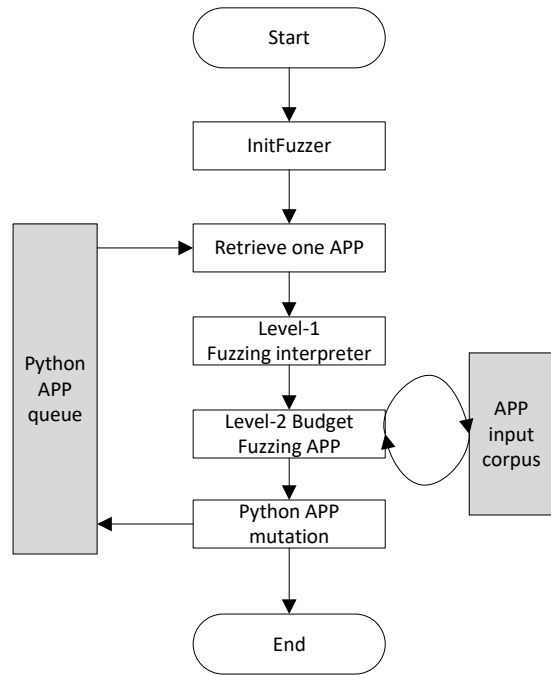
Software design



Example



Two-level Fuzzing



Level-1 Fuzzing: Targeting interpreter Core. We use infinite budget at level-1.

Level-2 Fuzzing: Targeting runtime libraries. We use finite budget at level-2 until no favored path/block/feature found.

```
<apisepc>
<library email>
  <module charset>
    <class Charset>
      <api>
        <name>get_body_encoding</name>
        <expr>ret = get_body_encoding()</expr>
        <parameters>{}</parameters>
        <return>{'ret':base64|7bit}</return>
        <dependencies> </dependencies>
      </api>

      <api>
        <name>header_encode</name>
        <expr>header_encode(str)</expr>
        <parameters>{'str':string}</parameters>
        <return>{}</return>
        <dependencies> </dependencies>
      </api>
    </class>
  </module>

  <module contentmanager>
    <api>
      <name>header_encode</name>
      <expr>header_encode(str)</expr>
      <parameters>{'str':string}</parameters>
      <return>{}</return>
      <dependencies> </dependencies>
    </api>
  </module >

  <errors>
    <exception>MessageError</exception>
    <exception>MessageParseError</exception>
    <exception>BoundaryError</exception>
    <exception>MultipartConversionError</exception>
  </errors>
</library>
</apisepc>
```

Object-oriented

```
class demoCls:
    def __init__(self):
        pass
    def demoFunc0(self):
        pass
    def demoFunc1(self, arg1):
        pass

def RunFuzzer (x):
    pass
```

Procedure-oriented

```
class demoCls:
    def __init__(self):
        pass
    def demoFunc0(self):
        pass
    def demoFunc1(self, arg1):
        pass

def RunFuzzer (x):
    pass
```

Program structure

```
class demoCls (PY_LIB.CLS):
    def __init__(self):
        pass
    def CLS_func0(self):
        pass
    def CLS_func1(self, arg1):
        pass

def RunFuzzer (x):
    pass
```

```
For intV in range (x, y):
    pass
```

```
For obj in iterObj:
    pass
```

Function structure

```
if x:
    pass
```

```
if x:
    pass
Else:
    pass
```