

**KAMAILIO
CONF/25**

Маршрутизация вызовов в Kamailio





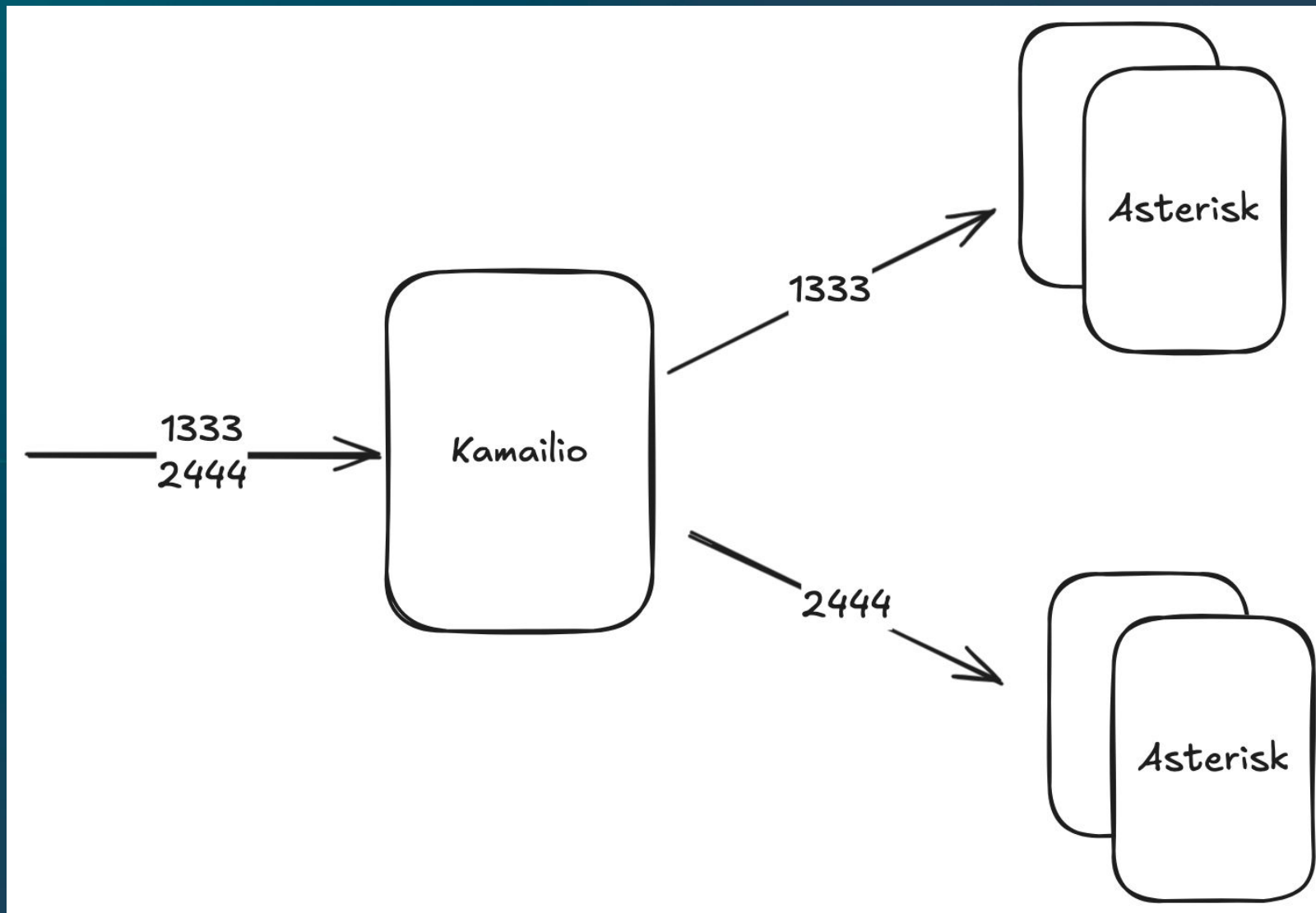
**Евгений
Тевелевич**

Проблематика доклада:



- Маршрутизация вызовов на основе вызываемого номера
- Минимум логики на стороне Kamailio
- Использование модулей dialplan и dispatcher

Схема



Не будет в докладе



- **Вариантов маршрутизации** - обзор и сравнение
- **Модуль dispatcher** - настройка и принцип работы
- **Модуль dialplan** - разбор сложных случаев: dynamic, branch
- **Regex** - как работают

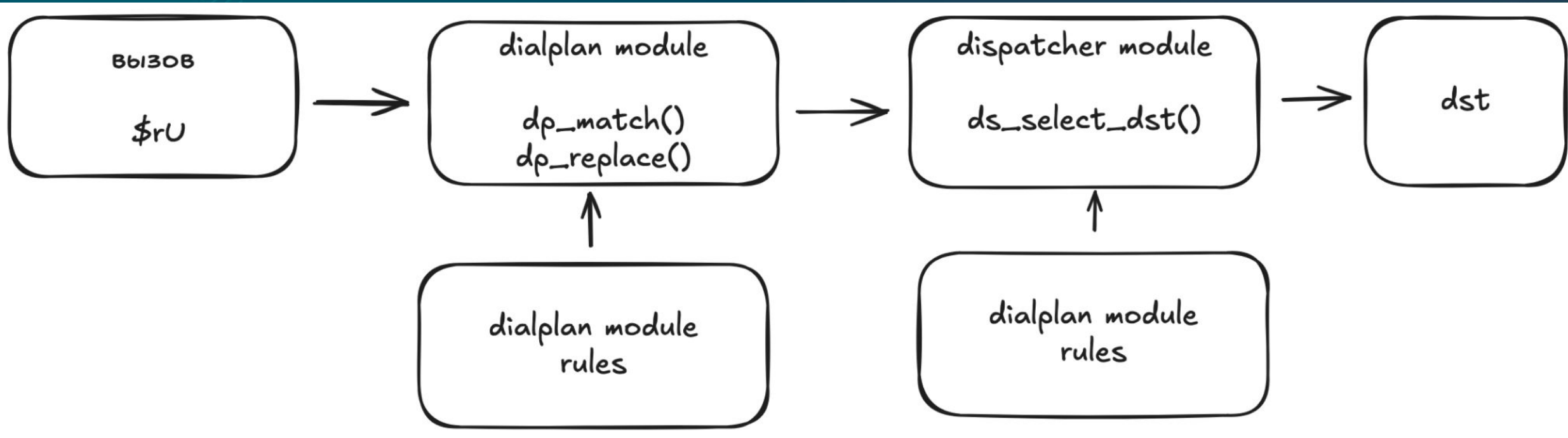


Что будет



- **Модуль dialplan** - назначение, настройка, применение
- **функции модуля** - чем отличаются, примеры
- **Kamailio vs Opensips** - отличия в реализации модуля dialplan
- **debug**

Основная идея



dialplan - принцип работы модуля



- При запуске/по команде Kamailio модуль загружает правила из базы данных
- Правила группируются в диалпланы по идентификатору `dpid`
- При обработке вызова входное значение (номер) сопоставляется с правилами
- Используются методы сопоставления: регулярные выражения, строковое сравнение и длина строки
- При совпадении применяется преобразование и возвращаются атрибуты

dialplan - параметры модуля



- **attrs_pvar(string)** - определяет имя псевдопеременной, которая заполняется из поля attrs
- **fetch_rows (int)** - ограничивает кол-во строк которое будет прочитано за одно обращение к БД
- **match_dynamic (int)** - предполагает, что выражения могут содержать переменные и выражение будет вычисляться во время выполнения.
- **append_branch (int)** - позволяет делать branch для случая если сработает сразу несколько совпадение
- **reload_delta (int)** - позволяет ограничить частоту загрузки из БД.

dialplan - поля таблицы БД



- **dpid** – идентификатор диалплана (позволяет иметь несколько независимых планов набора).
- **pr** – приоритет правила (определяет порядок проверки правил).
- **match_op** – тип операции сравнения (сравнение строк: 0, regexp: 1).
- **match_exp** – выражение для сопоставления (строка или regexp).
- **match_len** – длина строки (0 - любая длина, влияет на порядок сопоставления).
- **subst_exp** – регулярное выражение для подстановки..
- **repl_exp** – результат замены или выражение для формирования рез-та замены..
- **attrs** – дополнительные атрибуты (метки, параметры для логики маршрутизации).

dialplan - пример таблицы



id	dpid	pr	match_op	match_exp	match_len	subst_exp	repl_exp	attrs
28	1	10	1	^1\d{3}\$	0		1	dispatcher_id=1
29	1	10	1	^2\d{3}\$	0		2	dispatcher_id=2

id	dpid	pr	match_op	match_exp	match_len	subst_exp	repl_exp	attrs
30	2	10	1	^1234\$	0		1	dispatcher_id=1
31	2	10	0	1234	0		2	dispatcher_id=2
32	2	10	1	1234	4		3	dispatcher_id=3
33	2	10	1	1234	0		4	dispatcher_id=4

dialplan - функции модуля



dp_replace("dpid", "inval", "outvar")

или

dp_translate("dpid", "inval/outvar")

Переводит входное значение **inval** с использованием правил диалплана **dpid**.

Результат сохраняется в переменной **outvar**.



dp_match("dpid", "inval")

или

dp_translate("dpid", "inval")

Сопоставляет входное значение **inval** с правилами диалплана **dpid**.



Атрибуты сохраняются в переменной определенной параметром модуля **attrs_pvar**

Возвращает: 1 (успех), -1 (ошибка), -2 (диалплан не существует)

dialplan - команды загрузки, отладки



- **kamcmd dialplan.reload** (загружает правила из БД).
- **kamcmd dialplan.dump** (выводит действующие правила для указанного dpid в отсортированном по приоритету порядке).
- **match_op** – тип операции сравнения (сравнение строк: 0, regexp: 1).



```
kamcmd dialplan.translate 1 s:1333
{
    Output: 1
    Attributes: dispatcher_id=1
}
```

```
kamcmd dialplan.translate 1 s:2333
{
    Output: 2
    Attributes: dispatcher_id=2
}
```

```
kamcmd dialplan.dump 1
{
    DPID: 1
    ENTRIES: {
        ENTRY: {
            PRI0: 10
            MATCHOP: 1
            MATCHEXP: ^1\d{3}$
            MATCHLEN: 0
            SUBSTEXP: <null string>
            REPLEXP: 1
            ATTRS: dispatcher_id=1
        }
        ENTRY: {
            PRI0: 10
            MATCHOP: 1
            MATCHEXP: ^2\d{3}$
            MATCHLEN: 0
            SUBSTEXP: <null string>
            REPLEXP: 2
            ATTRS: dispatcher_id=2
        }
    }
}
```



```
kamcmd dialplan.translate 2 s:1234
{
    Output: 3
    Attributes: dispatcher_id=3
}
```

```
kamcmd dialplan.translate 2 s:12345
{
    Output: 4
    Attributes: dispatcher_id=4
}
```

```
kamcmd dialplan.reload;kamcmd dialplan.dump 2
{
    DPID: 2
    ENTRIES: {
        ENTRY: {
            MATCHOP: 1
            MATCHEXP: 1234
            MATCHLEN: 4
            ATTRS: dispatcher_id=3
        }
        ENTRY: {
            MATCHOP: 1
            MATCHEXP: ^1234$
            MATCHLEN: 0
            ATTRS: dispatcher_id=1
        }
        ENTRY: {
            MATCHOP: 0
            MATCHEXP: 1234
            MATCHLEN: 0
            ATTRS: dispatcher_id=2
        }
        ENTRY: {
            MATCHOP: 1
            MATCHEXP: 1234
            MATCHLEN: 0
            ATTRS: dispatcher_id=4
        }
    }
}
```

если приоритет(pr) равен?



- **match_len != 0** приоритетнее **match_len = 0**
- **чем меньше значение pr тем выше приоритет**
- **чем меньше значение id (порядок вставки) тем выше приоритет**

kamailio vs opensips



- только функция `dp_translate()`
- изменен синтаксис, выходное значение как отдельный параметр
- значение поля `attrs` передается явно через параметр функции
- отсутствует возможность сопоставление по длине строки
- и др. отличия

KAMAILIO
CONF/25

Спасибо за внимание!

Буду рад ответить на все ваши вопросы
сейчас или свяжитесь со мной в будущем:



Евгений Тевелевич

eyt5297@gmail.com

<tg://eyt5297>