

The background features abstract geometric shapes in various shades of blue, including triangles and polygons, creating a modern and dynamic visual effect.

# Інформаційні системи Лабораторна робота 4

Викладач: к.т.н., доц. Саяпіна Інна Олександрівна

# Порядок встановлення RabbitMQ

- ▶ 1. Встановлюємо мову Erlang, на якій написана програма - <https://www.erlang.org/>
- ▶ 2. На сайті <https://www.rabbitmq.com/download.html> скачуємо відповідну версію
- ▶ 3. У брендмауері дозволяємо з'єднання (порти 5672, 15672)
- ▶ 4. Для увімкнення плагіну переходимо на сторінку управління <https://www.rabbitmq.com/management.html> . Відкриваємо командний рядок та допомогою команди у **rabbitmq-plugins enable rabbitmq\_management** з папки установки (зазвичай C:\Program Files\RabbitMQ Server\rabbitmq\_server-3.11.13\sbin) запускаємо керуючі плагіни.
- ▶ 5. З браузера заходимо на <http://localhost:15672/>
- ▶ Якщо сторінка не вантажиться перезавантажуємо комп'ютер
- ▶

# Порядок встановлення RabbitMQ

- ▶ За адресою <http://localhost:15672/> відкривається вікно



The image shows the RabbitMQ management interface login page. At the top is the RabbitMQ logo, which consists of an orange icon of three stacked squares followed by the text "RabbitMQ" in orange and "TM" in grey. Below the logo are two input fields: "Username:" and "Password:". Each field has a small red asterisk to its right. Below the password field is a dark grey button with the word "Login" in white text.

- ▶ Логін: guest
- ▶ Пароль: guest

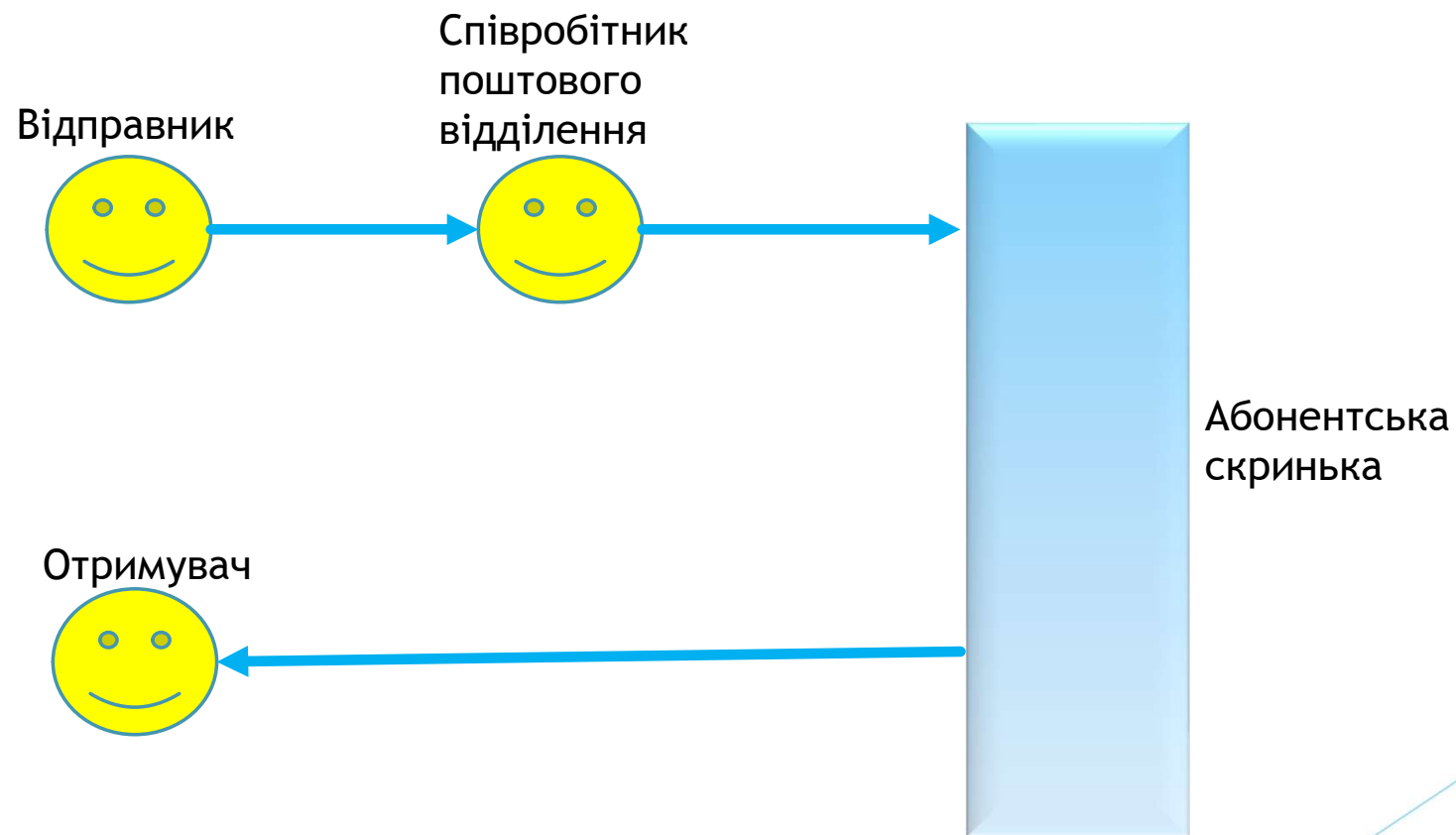
# Порядок встановлення RabbitMQ

- ▶ Якщо сторінка все одно не завантажується пробуємо наступні дії:
  - 1 Запускаємо командний рядок від імені адміністратора
  - 2 Виконуємо команду `C:\Program Files\RabbitMQ Server\rabbitmq_server-YOUR_VERSION\sbin>rabbitmqctl.bat stop`
  - 3 Виконуємо команду `C:\Program Files\RabbitMQ Server\rabbitmq_server-YOUR_VERSION\sbin>rabbitmqctl.bat start_app`

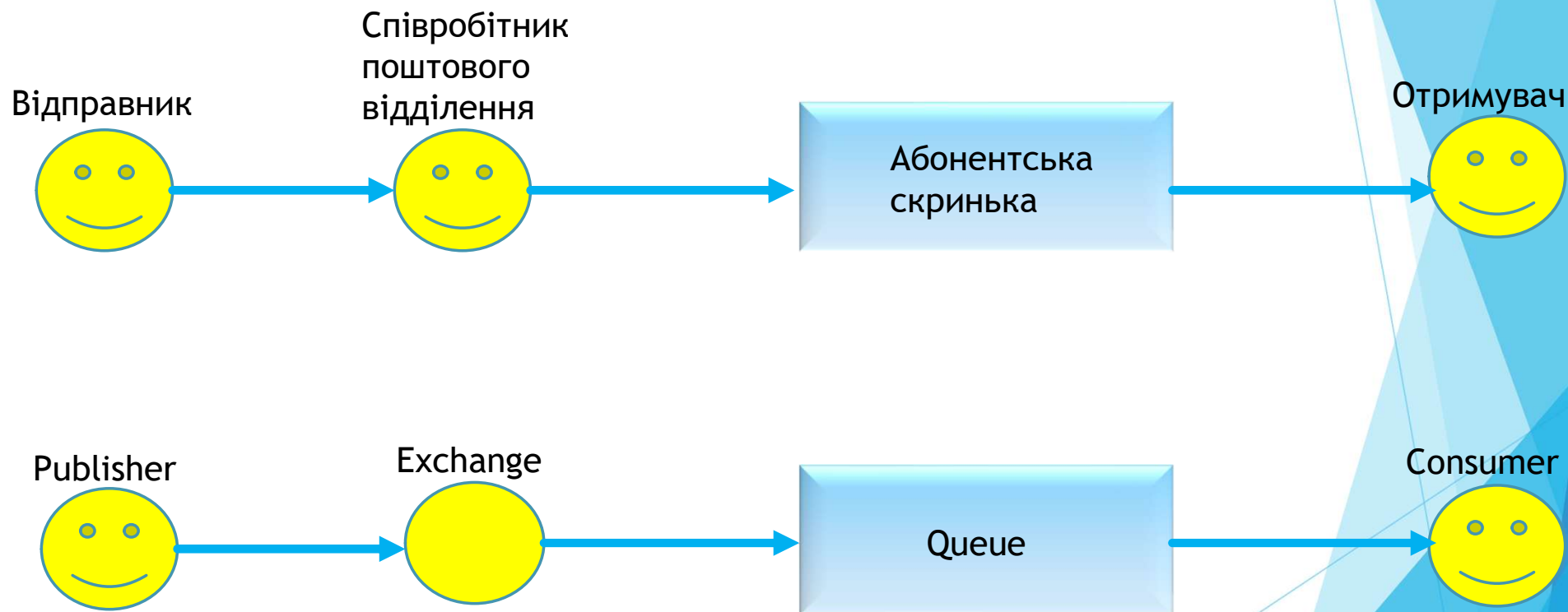
Або

- ▶ 1 Натискаємо кнопку Пуск
- ▶ 2 Пишемо Services та натискаємо Enter
- ▶ 3 У вікні, що відкрилось, знаходимо сервіс з назвою RabbitMQ та запускаємо його

# Принцип роботи RabbitMQ

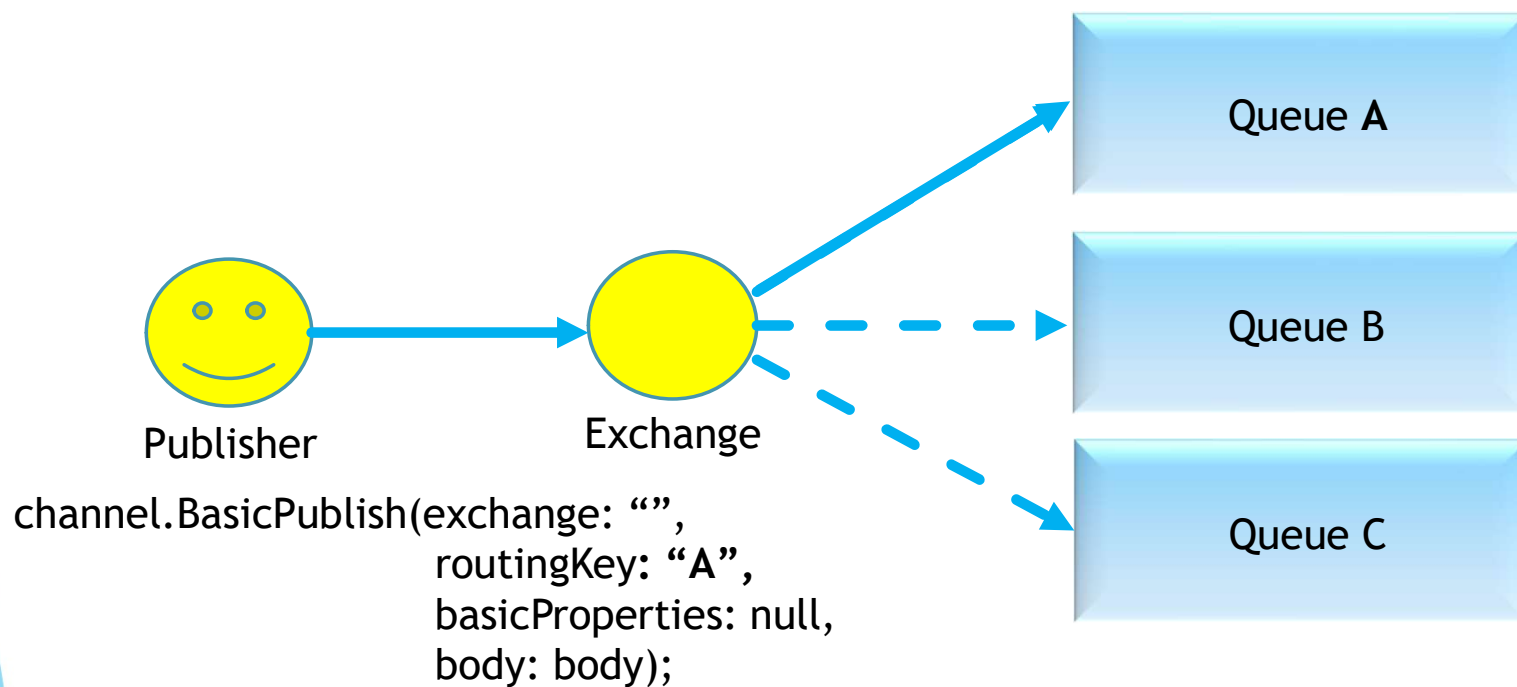


# Принцип роботи RabbitMQ



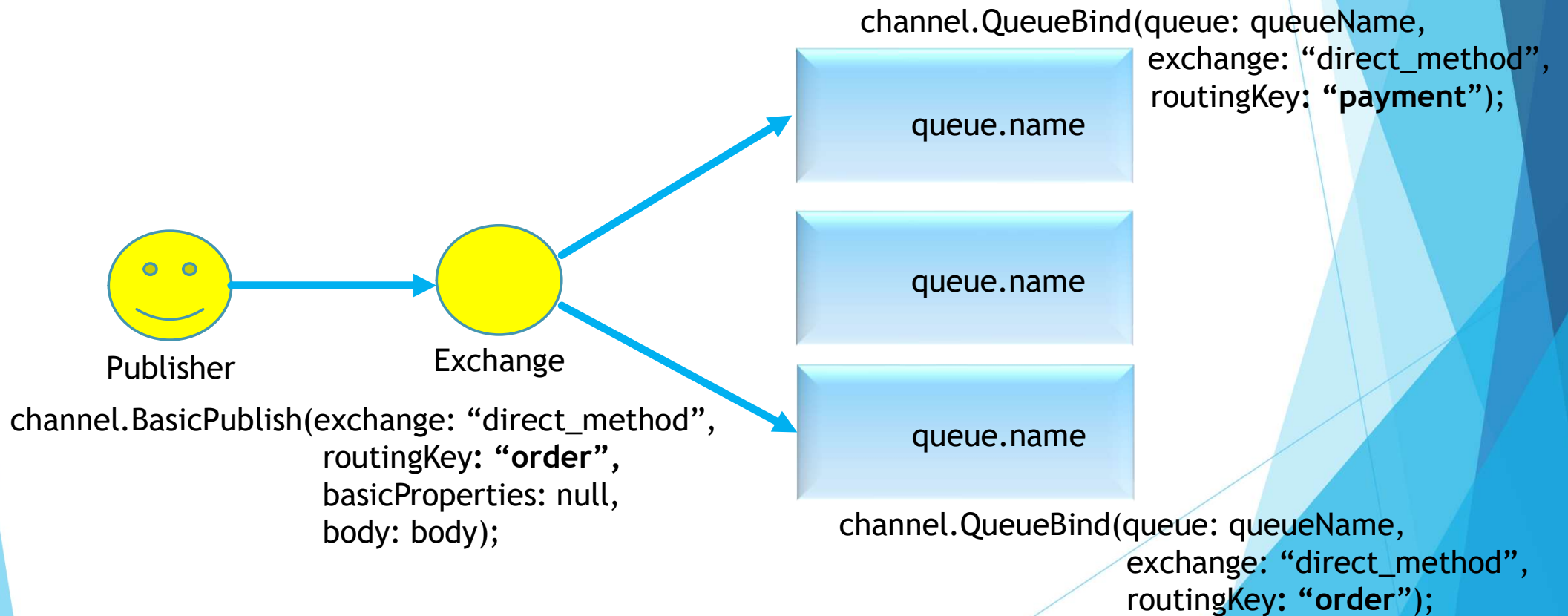
# Default Exchange

- Default Exchange - не пов'язаний не з однією чергою та відправляє повідомлення у будь-яку з черг, назва якої співпадає з Routing key



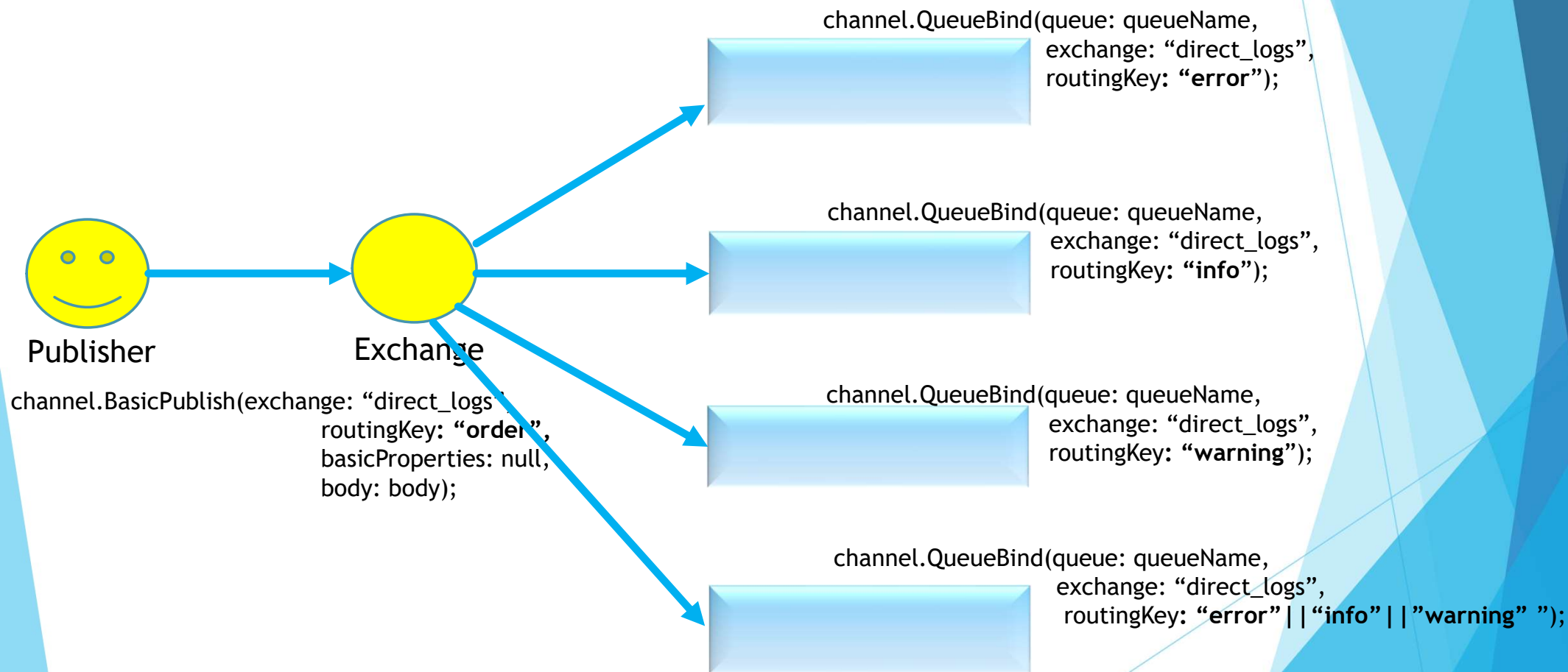
# Direct Exchange

- **Direct Exchange** - має набір зв'язків (bindings) та пов'язаний з певними чергами, а повідомлення буде спрямовано у ту чергу, у якої співпадає параметр Routing key і немає різниці яка назва черги.



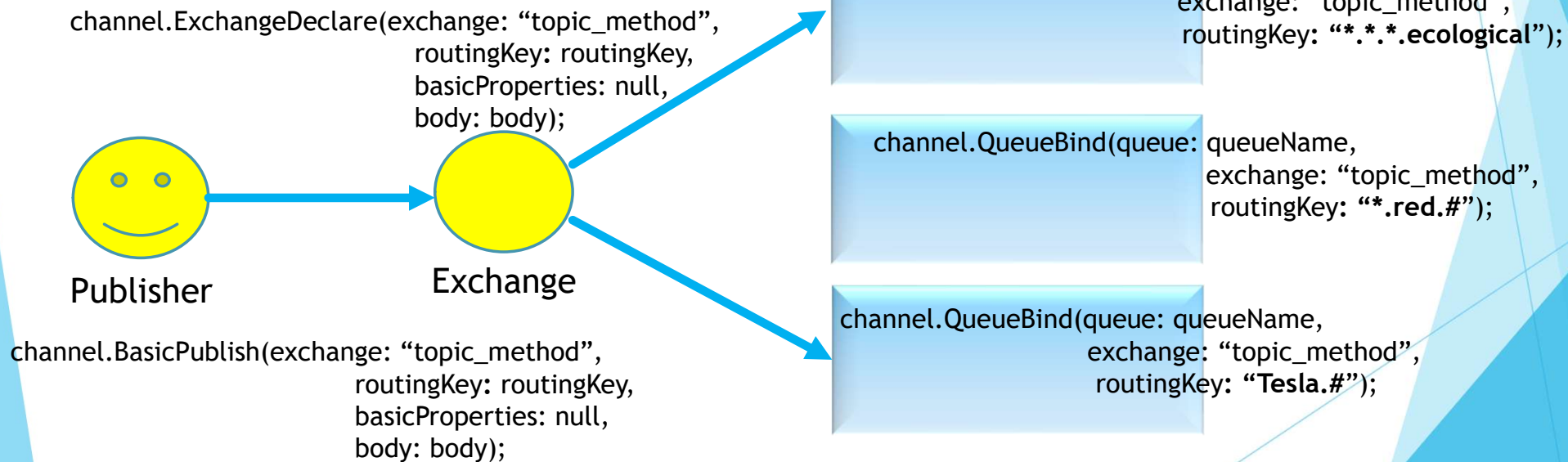


# Приклад реалізації Direct Exchange



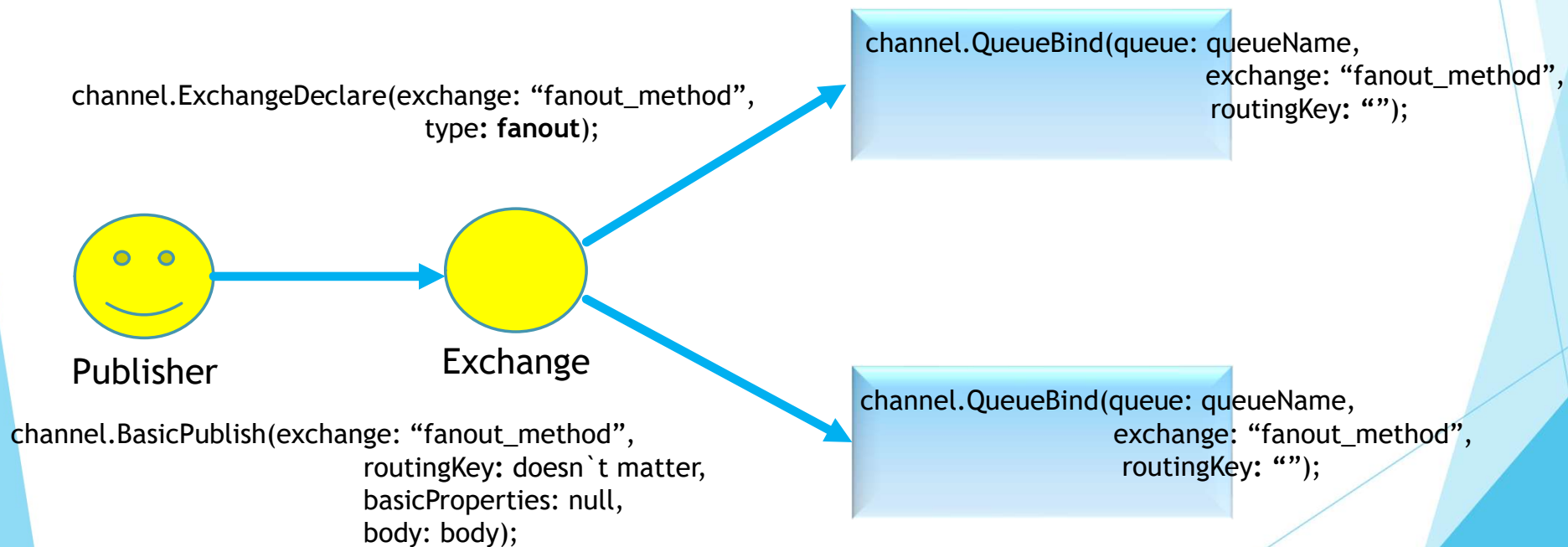
# Topic Exchange

- ▶ **Topic Exchange** - обмінник, що здійснює маршрутизацію на основі одразу декількох критеріїв. У ролі критеріїв виступають слова та способи прив'язки за цими словами. Маршрутизація здійснюється за параметром `RoutingKey`, але тут він представляє собою набір слів, розділених крапкою.
- ▶ \* - будь-яке слово (`*.*.*.ecological=слово.слово.слово.ecological`)
- ▶ # - нуль чи більше слів (`red.#= red.слово.слово...`)



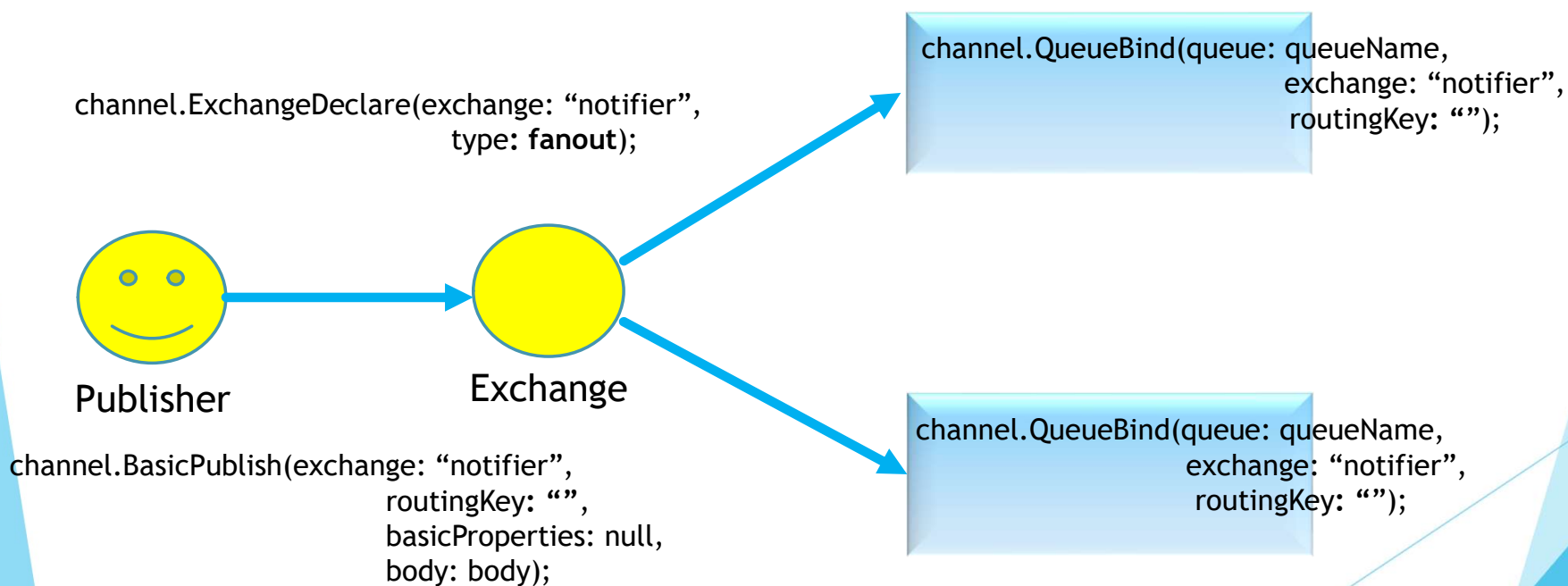
# Fanout Exchange

- **Fanout Exchange** - обмінник, що здійснює маршрутизацію відправляє повідомлення у всі черги, про які знає. На основі його реалізується патерн Publish/Subscribe.



# Приклад Fanout Exchange

- Розберемо події надходження коштів на банківський рахунок та повідомлення про це сервіс відрахувань для податкової (Consumer Tax), сервіс зберігання подій про зарахування коштів на рахунок (Consumer BankCharger), та сервіс SMS оповіщення клієнта.



# Headers Exchange

- ▶ **Headers Exchange** - обмінник, що здійснює маршрутизацію повідомлень не на основі Routing Key, а на основі атрибутів заголовка.
- ▶ Приклад: Три сервіси, які отримують повідомлення і підраховують кількість подій переказу доларів за кордон, доларів всередині країни та переказу євро всередині країни.

