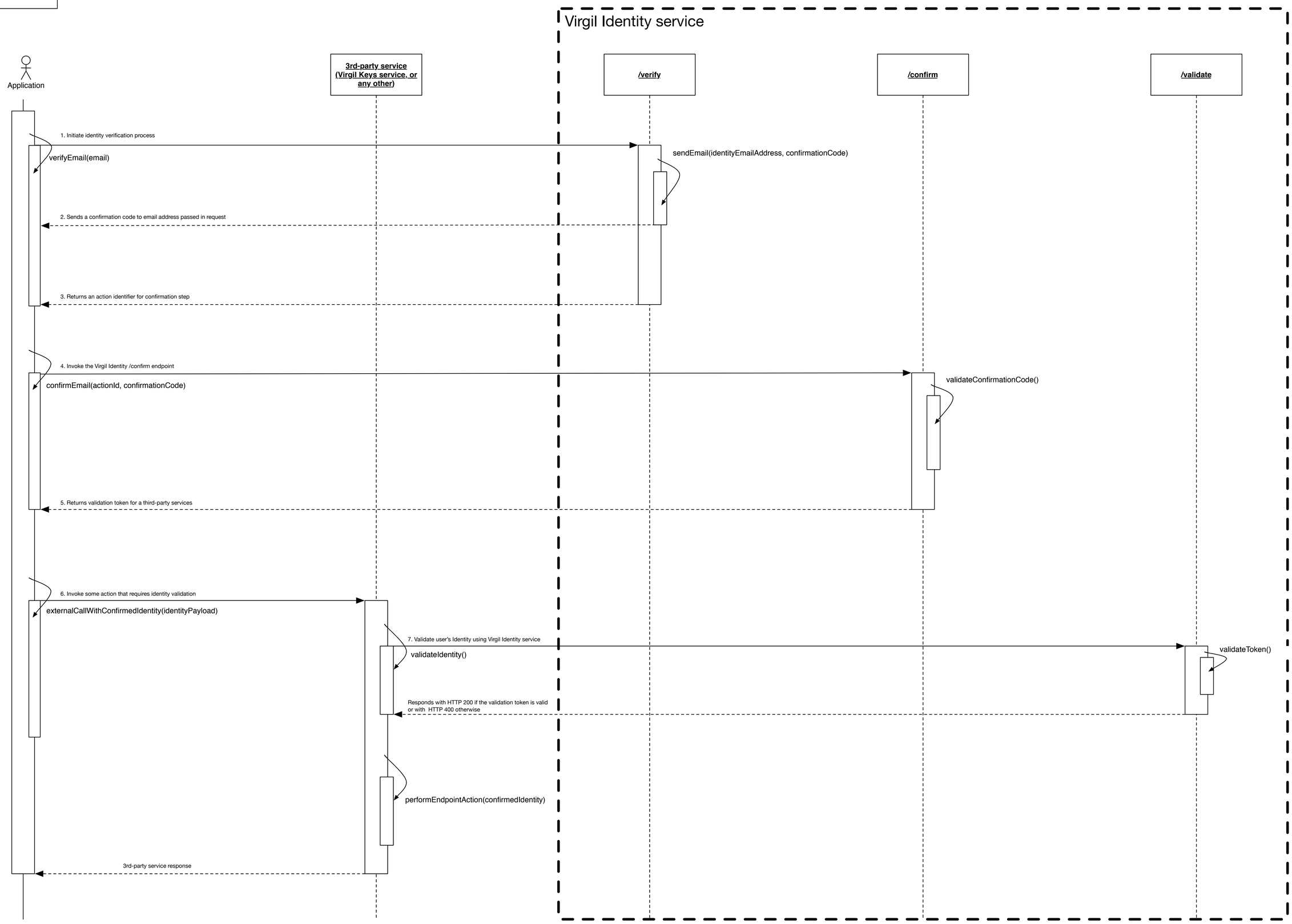
3.2. Virgil Identity. Sequence diagram



Legend

1. Initiates an Identity verification process. As a first step it invokes the Virgil Identity's /verify endpoint with request payload:

```
{
    "type": "email",
    "value": "user@virgilsecurity.com"
}
```

The Virgil Identity service initiates an Identity verification process and sends an email to the address specified in the request payload.

2. Virgil Identity service sends an email to the address specified in the request payload to make sure that user controls the claimed Identity. An email content contains the confirmation code like:

```
{ "confirmation_code": "G9K5M2" }
```

3. Virgil Identity /verify endpoint generates an action identifies to be used along with the confirmation code to make sure the user who received the confirmation token initiated the verify process. The response payload looks like:

```
{
    "action_id": "3ba5ccab-66f2-4422-bf17-b122b13a1edd"
}
```

4. Confirm an Identity by Virgil Identity /confirm endpoint invocation with request payload that contains a confirmation code from the email and action identifier that was returned in the response:

```
{
    "action_id": "3ba5ccab-66f2-4422-bf17-b122b13a1edd",
    "confirmation_code": "G9K5M2"
}
```

5. Virgil Identity service verifies that action_id, confirmation_code pair. It returns a validation token value if the confirmation code is correct. The validation token is used on the validation endpoint to prove that the user passed the process of Identity verification. The response format is:

```
{
    "type": "email",
    "value": "user@virgilsecurity.com",
    "validation_token": "MIIB5wIBADCCAeAGCSqGSIb3…"
}
```

6. To invoke the 3rd-party service that uses Virgil Identity service as an Identity verification engine, it's necessary to pass the whole response from the step #5 to the external service in order to perform Identity validation and make sure that current user controls the claimed Identity. The possible request format is format is:

```
{
    …
    "identity": {
        "type": "email",
        "value": "user@virgilsecurity.com",
        "validation_token": "MIIB5wIBADCCAeAGCSqGSIb3…"
    }
}
```

7. The 3rd-party service has the possibility to make sure that the user is the valid Identity holder by an invocation of the /validate endpoint. A request payload must be the same as a response from the step #5. It's not necessary to use this scheme with 3rd-party services, it's possible to invoke the /validate endpoint in the application itself to make sure that a user is a valid Identity holder. The request format is:

```
{
    "type": "email",
    "value": "user@virgilsecurity.com",
    "validation_token": "MIIB5wIBADCCAeAGCSqGSIb3…"
}
```