4.0 SmartLink

4.1 Introduction

The SmartLink technology allows us to share with a pet owner an individualised link to the booking system, which when clicked allows the pet owner to make a booking without entering any personal information. SmartLink technology works by packaging some client specific information into the URL itself, so that when the URL is accessed, Vetstoria is able to see which person the request belongs to.

This technology is designed to be utilised by the PMS when sending reminders to pet owners, where a smart link can be embedded in each reminder, either in the email or in an sms, so that when clicked, it will take the pet owner directly to the booking system, with all personal information pre-filled. Depending on the level of information embedded in the link the booking process could be as simple two-steps.

4.1 Specification

Basic URL Format:
<protocol>://<base_url>/<site_specific_hash>/<encrypted_booking_data>

Responsibility:

<protocol>	: provided by Vetstoria - normally HTTPS
<base_url>	: provided by Vetstoria. Will normally be uniform for clinics in a single region.
<site_specific_hash>	: provided by Vetstoria. This will change for each site/clinic.
<payload>	: dynamically generated and appended to the URL.

Booking Data:
Vetstoria will expect the <encrypted_booking_data> component of the URL to be a base64 encoded JSON object encrypted using an agreeable encryption protocol. It will typically contain the following attributes. Extensibility should be allowed for additional data attributes in the future.

"c_id" - client id from the PMS
"pets" - a list of detail blocks, one for each pet, containing attributes p_id, at_id, rfa_id & rfa_text
    "p_id" - pet id from the PMS
    "at_id" - appointment type id (optional)
    "rfa_id" - reason for appointment id (optional)
    "rfa_txt" - reason for appointment text (optional)
"rpd" - 16 characters of random padding data - can be any random 16 characters

Example JSON Structure:

```
{
    "c_id": "1",
    "pets": [
        {
            "p_id": "2",
            "at_id": "3",
            "rfa_id": "77"
        },
        {
            "p_id": "3",
            "rfa_id": "72"
        }
    ],
    "rpd": "RxuagFJNJRf8aS6M"
}
```

Encoding and Encryption Steps:

Data = '`{"c_id": "1","pets": [{"p_id": "2","at_id": "3","rfa_id": "77"}, {"p_id": "3","rfa_id": "72"}],"rpd": "RxuagFJNJRf8aS6M"}`';
encoded_data = base64encode(data);
encrypted_data = encrypt(encoded_data, shared_encryption_key);
payload = urlsafe_base64encode(encrypted_data);

***The encryption algorithm is to be agreed between both parties before implementation.**

Example URL:
https://appointments.vetstoria.com/52b1d902a3f/eyJhh7IjoxLCJjIjozfefoefLCJyItptmvfk8u378f39fj3joibm8gcmVhc29uIn0

SMS Embedding:

The URL may be shortened for SMS/Email using a service such as the google url shortener. https://developers.google.com/url-shortener/v1/getting_started