

Supplementary Materials

Displacing Black-legged Kittiwakes *Rissa tridactyla* – First experiences from the Kittiwake hotel in Røst, northern Norway

Tycho Anker-Nilssen^{1*}, Lisa Alcaraz², Soheyra Anjomipoor³, Aline Magdalena Lee^{3,4}, Thor Harald Ringsby³ and Annette L. Fayet^{1*}

Timeline illustrations and technical details

List of contents

- Figure S1 – June 2018: Close-up of the colony on Nilssenhuset in its peak year.
- Figure S2 – May 2020: Nilssenhuset and the construction site before the Kittiwake hotel was built.
- Figure S3 – Spring 2023: Architectural drawings of exterior wall-faces of the Kittiwake hotel.
- Figure S4 – Spring 2023: Architectural drawings of main construction details of the Kittiwake hotel.
- Figure S5 – September 2023: Demounting and storage of nests from the old roof on Nilssenhuset.
- Figure S6 – October-November 2023: Raising the structural envelope of the Kittiwake hotel.
- Figure S7 – January 2024: Prefabricating nest platforms for the Kittiwake hotel.
- Figure S8 – February 2024: Mounting nest platforms on the Kittiwake hotel.
- Figure S9 – March 2024: Placing nests on the nest platforms at the Kittiwake hotel.
- Figure S10 – June 2026: Checking nest contents with an endoscope from inside the Kittiwake hotel.
- Figure S11 – The setup of control units inside the Kittiwake hotel.
- Table S1 – Python script for downloading CCTV time-lapse snapshots to an internet server.

Contacts for additional information:

Tycho Anker-Nilssen, Norwegian Institute for Nature Research, tycho@nina.no

Annette L. Fayet, Norwegian Institute for Nature Research, annette.fayet@nina.no



Figure S1. Part of the Black-legged Kittiwake *Rissa tridactyla* colony on Nilssenuset on 20 June 2018, when numbers on the roof peaked at 355 pairs, of which 240 pairs bred on the crowded, eastern side. © Tycho Anker-Nilssen.



Figure S2. View of Nilssenuset and the construction site on Kårøya on 13 May 2020. The Kittiwake hotel was built on the concrete walls in the foreground, which supported two large fuel tanks in the 1930s. The island in the background is the 205 m high Vedø, four km southwest of the harbour, which supported the large cliff-breeding population that became extinct in 2020. © Tycho Anker-Nilssen.

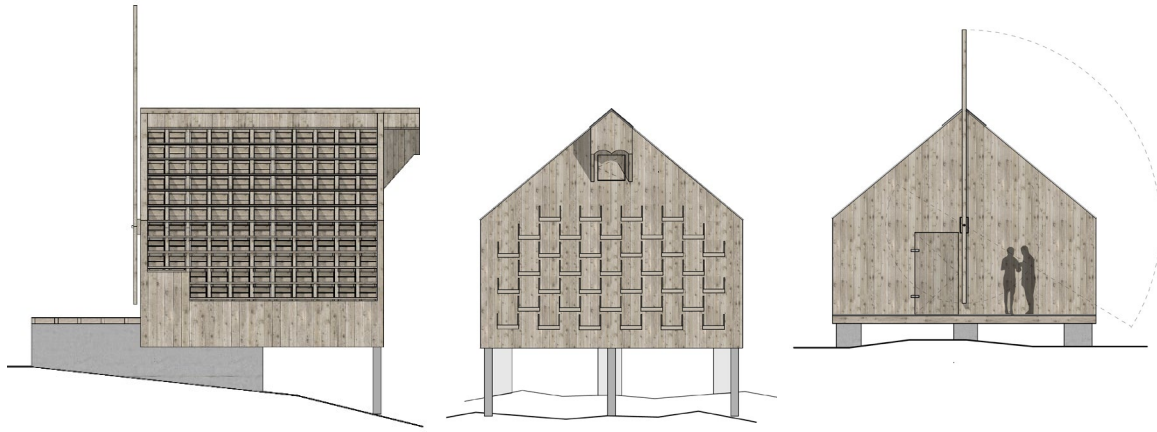


Figure S3. Exterior wall-faces of the Kittiwake hotel in Røst; the east-facing side wall (left), the north-facing facade (centre) and the south-facing entrance side. A CCTV camera is mounted on top of the tiltable mast (see Figure S10), and another inside the “takkelhus” on top of the seaside facade. *Architectural drawings: Kjetil Sæterdal; General design: Tycho Anker-Nilssen.*

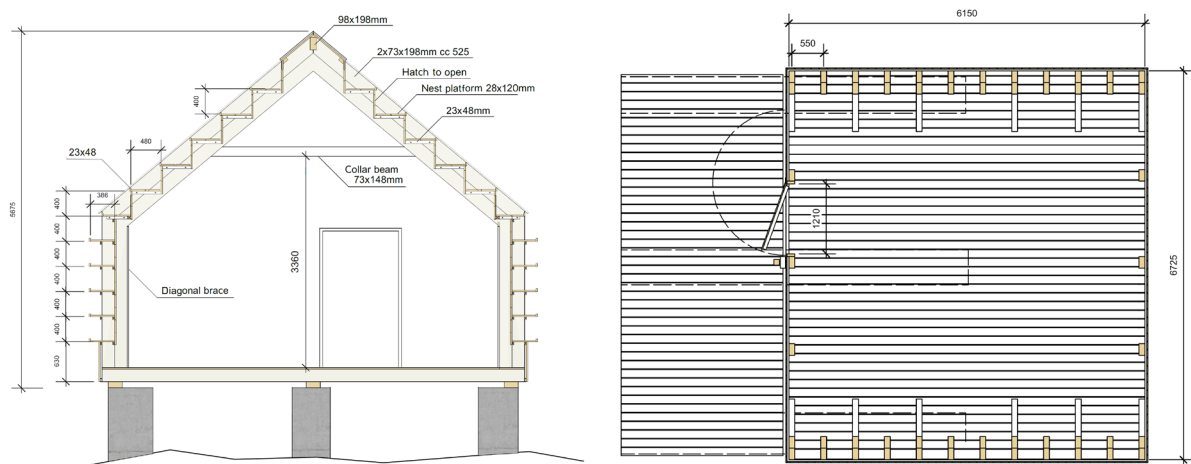


Figure S4. Construction details of the Kittiwake hotel in Røst: front-viewed cross section (left) and plan view (right). Dimensions of building materials are indicated. The final construction had 11 framework stands set 60 cm apart, rather than 12 stands set 55 cm apart. *Architectural drawings: Kjetil Sæterdal.*



Figure S5. On 19–20 September 2023, the nests on the old roof were carefully taken down, one by one (left, © Kjetil Sæterdal). National Trust of Norway architect Kjetil Sæterdal also assisted in the process (centre, © Tom Melby). The nests were stored overwinter in wooden fishing boxes (right, © Arnold Wilhelmsen).



Figure S6. Connecting supporting beams and raising the structural envelope of the Kittiwake hotel in October–November 2023, alongside repairs to the roof of Nilssenhuset (seen in the background on the lower panels) was also repaired. All framework was built using pressure-treated timber. © DSN Entreprenør AS, Svolvær.



Figure S7. In January 2024, the 210 nest platforms were constructed from 28×120 mm pressure-treated decking planks and prefabricated indoors using stainless steel decking screws prior to being fitted onto the structural envelope (right, see also Figures S8–S9). To allow easy access for measuring eggs and chicks, the removable upper half of the back wall was attached using two stainless steel machine screws and wing nuts.

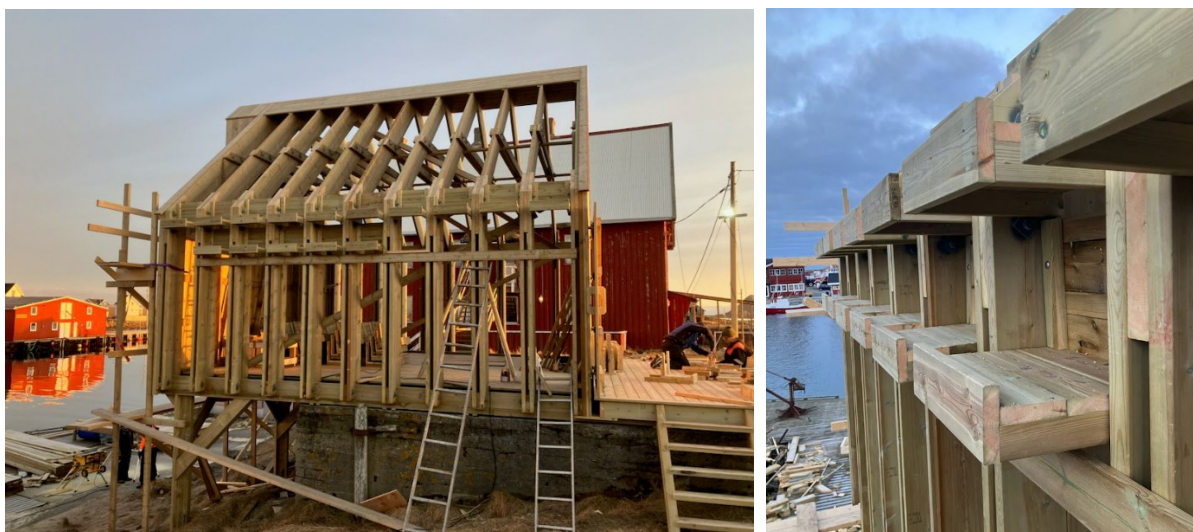


Figure S8. Diagonal braces were fitted on the inside of the walls to stiffen the building, and in January–February 2024, the prefabricated nest platforms were integrated into the structural envelope (© Arnold Wilhelmsen).



Figure S9. On 8 March 2024, the old nests from Nilssenhuset were mounted on the breeding platforms on the western side of the Kittiwake hotel, ten days later also on the eastern side of the roof (© Arnold Wilhelmsen).



Figure S10. Inspecting nest contents from inside the Kittiwake hotel through a small hole in the back wall of each nest platform, using a USB endoscope connected to a field computer (© Lisa Alcaraz).

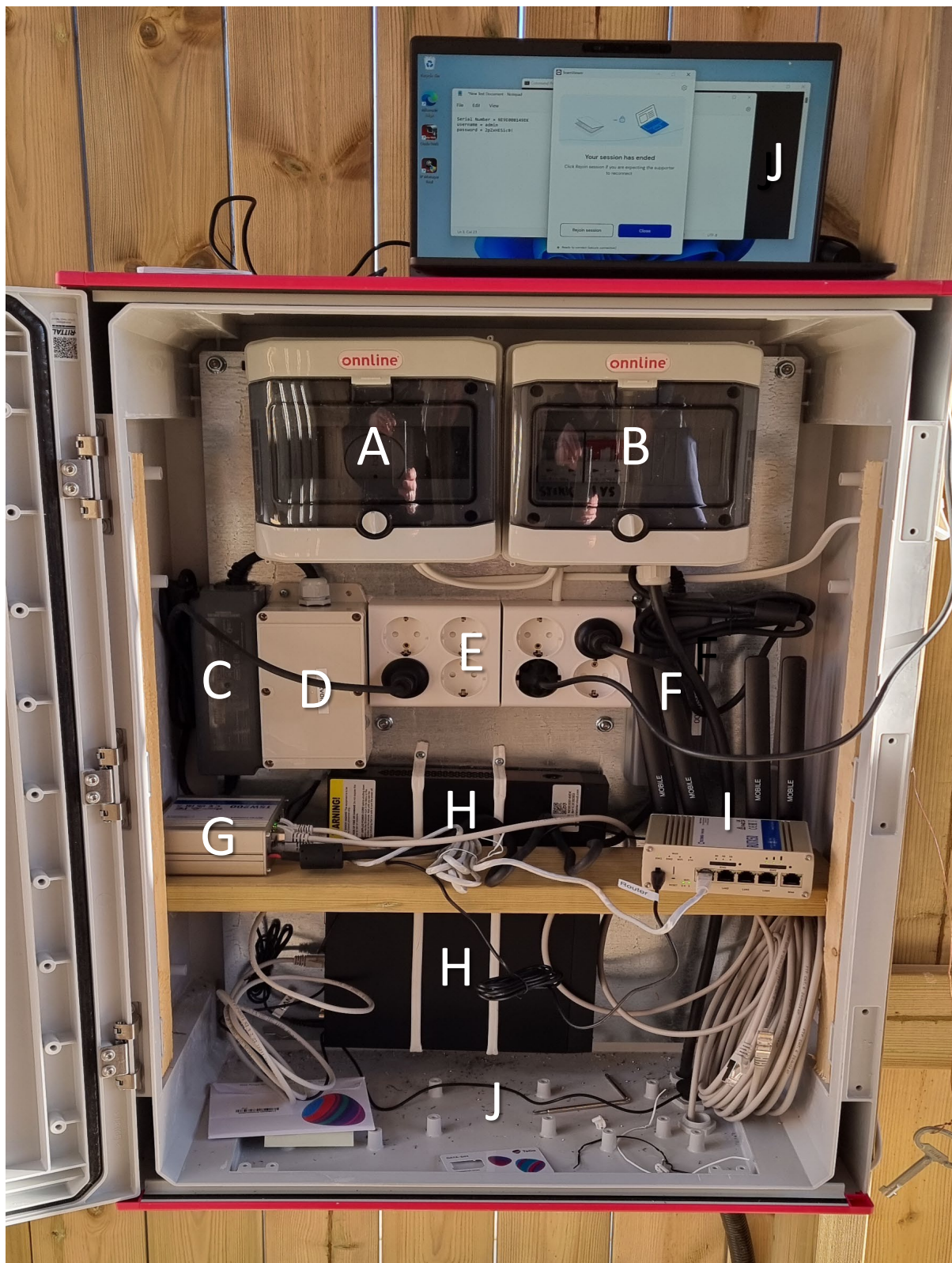


Figure S11. The setup of control units mounted in a waterproof cabinet inside the Kittiwake hotel: **A)** Timed switch for LED roof light, **B)** Mains 220V power switch & fuse, **C)** Power supply for the NVR (unit H), **D)** 12V DC converter, **E)** 230V/220V outlets, **F)** 48V DC converter for the **G)** Teltonika TSW200 Network switch with 8 RJ45 ports, **H)** Provision-ISR NVR8-8200N 8-channel network video recorder with 4TB HDD, **I)** Teltonika RUTX50 5G/LTE/3G router, **J)** The PC is shown pulled out for remote setup via TeamViewer (Version 15.x, TeamViewer Germany GmbH, Göppingen, Germany), but is otherwise stored at the cabinet base (© Tycho Anker-Nilssen).

Table S1. Anonymised version of the Python script used to set up automated downloading of time-lapse snapshot photos from the Provision-ISR CCTV cameras to an external web server (© *Kjetil Grun, NINA*).

```
#!/usr/bin/env python3
# encoding=utf-8

import argparse
import base64
import os
from datetime import datetime

import requests
from dotenv import load_dotenv

load_dotenv()

def get_arguments():
    """
    Parses command-line arguments and returns them as a dictionary.

    Returns:
    dict: A dictionary containing the parsed arguments.

    """
    arg_parser = argparse.ArgumentParser(
        description='Take snapshots from Provision-ISR IPC in required intervals',
        allow_abbrev=False
    )
    arg_parser.add_argument('-i', '--host', required=False,
                            default=os.environ.get('IPC_HOST'),
                            help='IP camera hostname or IP address.')
    arg_parser.add_argument('-p', '--port', type=int,
                            default=int(os.environ.get('IPC_PORT', 80)),
                            help='IPC Port.')
    arg_parser.add_argument('-u', '--username', type=str,
                            default=os.environ.get('IPC_USERNAME', 'admin'),
                            help='IPC Username.')
    arg_parser.add_argument('-P', '--password', type=str,
                            default=os.environ.get('IPC_PASSWORD'),
                            help='IPC Password.')
    arg_parser.add_argument('-n', '--name', type=str,
                            default=os.environ.get('SNAPSHOT_NAME', 'snapshot'),
                            help='Weather to overwrite the snapshot - use True if required')
    arg_parser.add_argument('-f', '--folder', type=str,
                            default=os.path.expanduser(os.environ.get('SNAPSHOT_FOLDER', '~')),
                            help='Folder to save the snapshots')
    args = arg_parser.parse_args()

    if not args.host:
        arg_parser.error("IPC_HOST must be set in .env or provided via --host")

    return vars(args)

def encode_credentials(username, password):
    """
    Encodes the username and password using base64 encoding.

    Args:
    username (str): The username.
    password (str): The password.

    Returns:
    str: The base64 encoded credentials.

    """
    encode_bytes = base64.b64encode((username + ':' + password).encode("utf-8"))
    encode_str = str(encode_bytes, "utf-8")
    return encode_str

def get_snapshot(args):
    """
    Downloads a snapshot from the camera and saves it to a file.

    Args:
    args (dict): The parsed command-line arguments.

    """
    headers = {
        'Authorization': 'Basic ' + encode_credentials(args['username'], args['password']),
        'Host': args['host']
    }
    API_ENDPOINT = 'http://' + args['host'] + ':' + str(args['port']) + '/GetSnapshot'

    try:
```

```

r = requests.get(API_ENDPOINT, headers=headers, timeout=10)

if r.status_code == 200:
    filename = args['name'] + "_" + datetime.now().strftime("%Y%m%d_%H%M%S") + ".jpg"
    filepath = os.path.join(args['folder'], filename)
    with open(filepath, "wb") as f:
        f.write(r.content)
    print(f"Snapshot saved as {filepath}")
else:
    print("Error downloading snapshot:", r.status_code)

except Exception as e:
    print("An error occurred:", e)

def main():
    """
    The main function of the script.
    """

    args = get_arguments()
    get_snapshot(args)

if __name__ == '__main__':
    main()

```